אני יודע מה עשית בפענוח האחרון: התקפות ערוצי צד על מחשבים אישיים

I Know What You Did Last Decryption: Side Channel Attacks on PCs

Daniel Genkin

Technion and Tel Aviv University

joint work with

Lev Pachmanov
Tel Aviv University

Itamar Pipman
Tel Aviv University

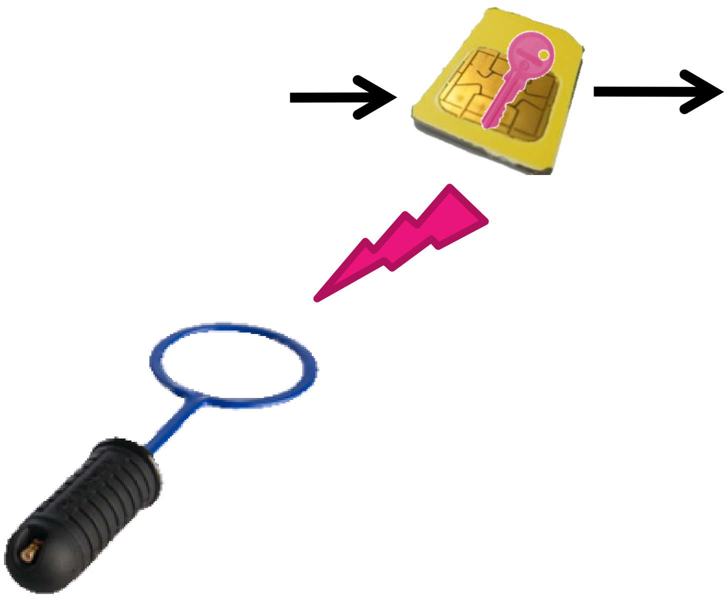
Adi Shamir
Weizmann Institute of Science

Eran Tromer
Tel Aviv University

Side channel attacks

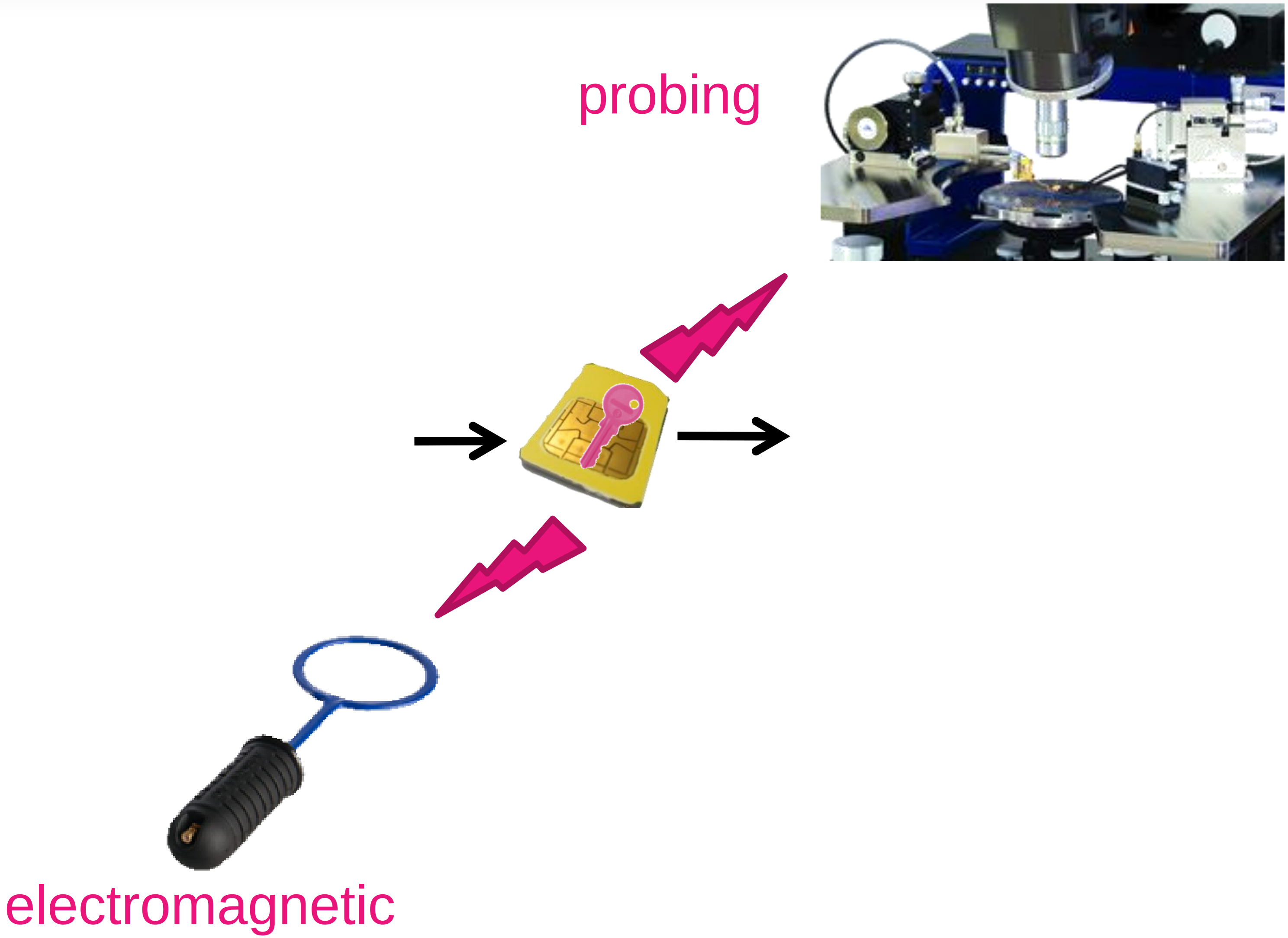
Side channel attacks

Side channel attacks

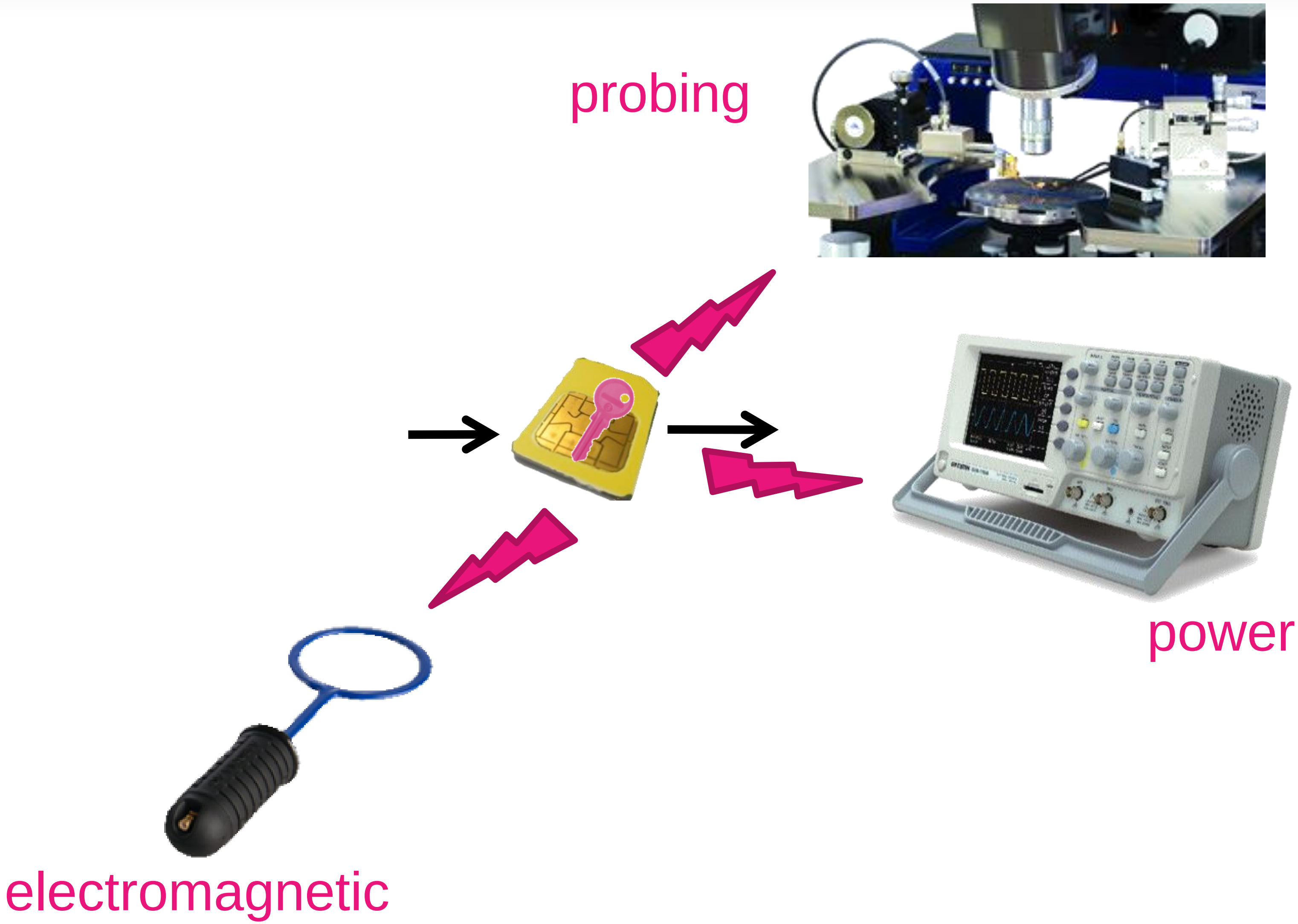


electromagnetic

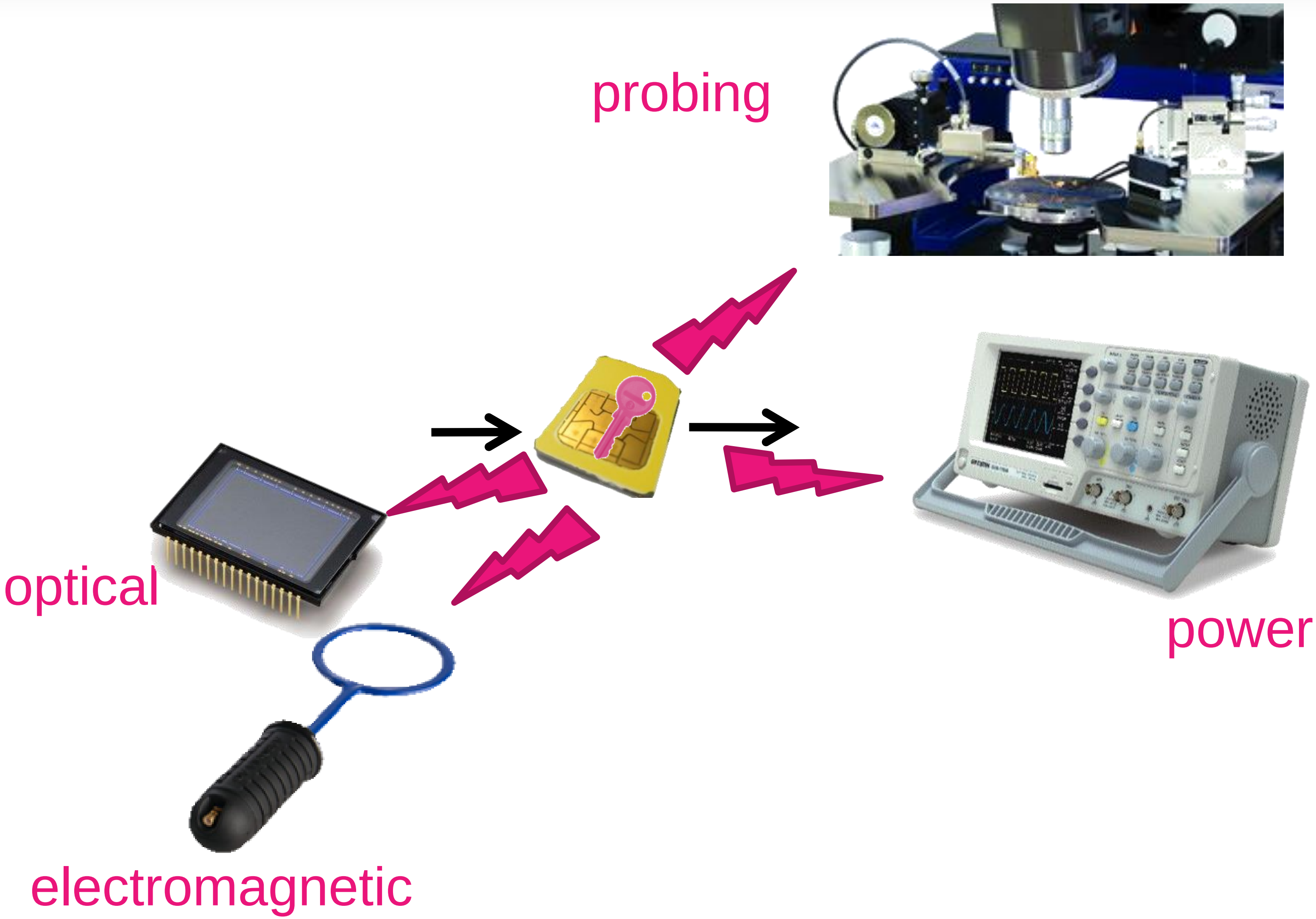
Side channel attacks



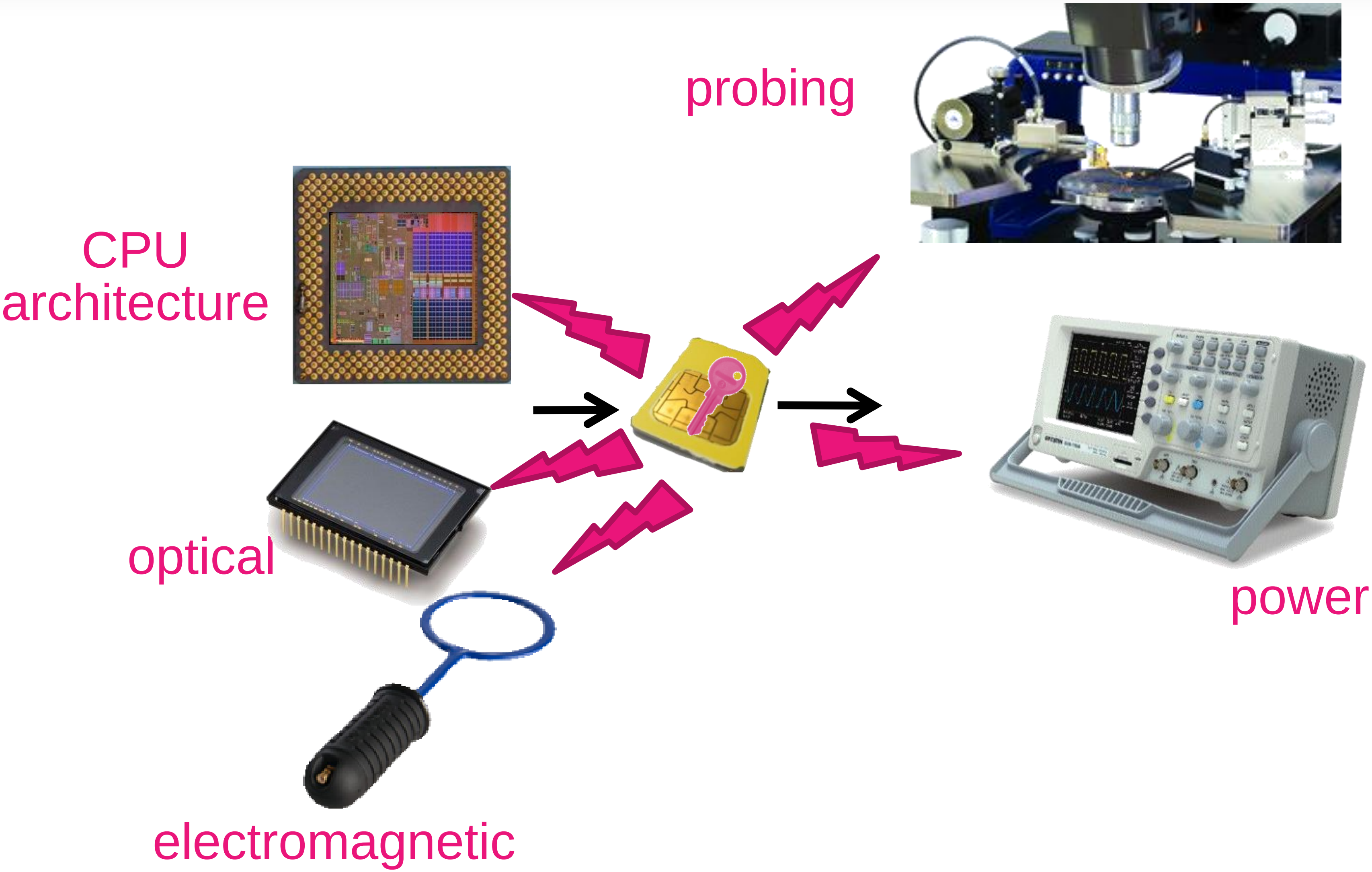
Side channel attacks



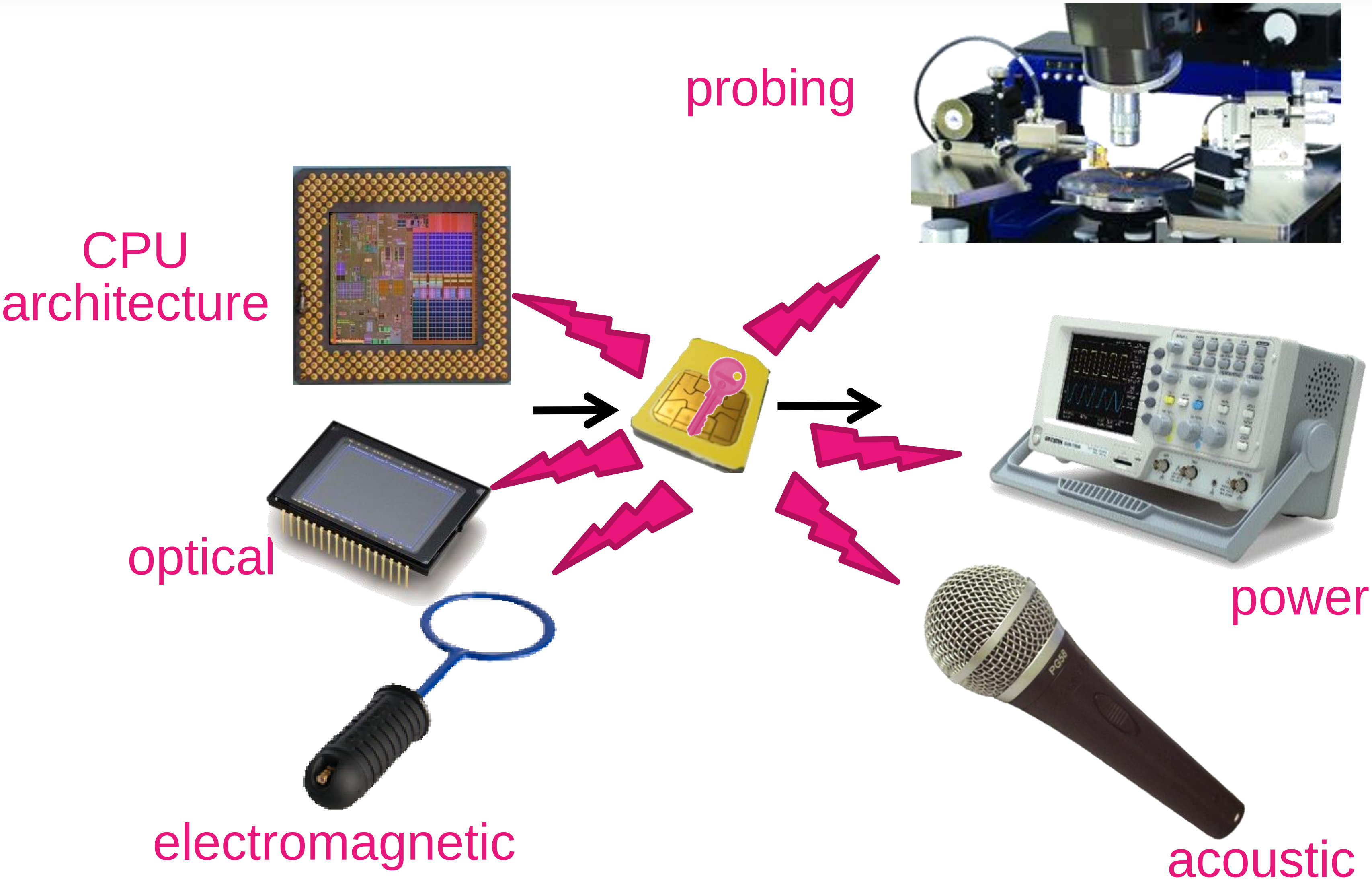
Side channel attacks



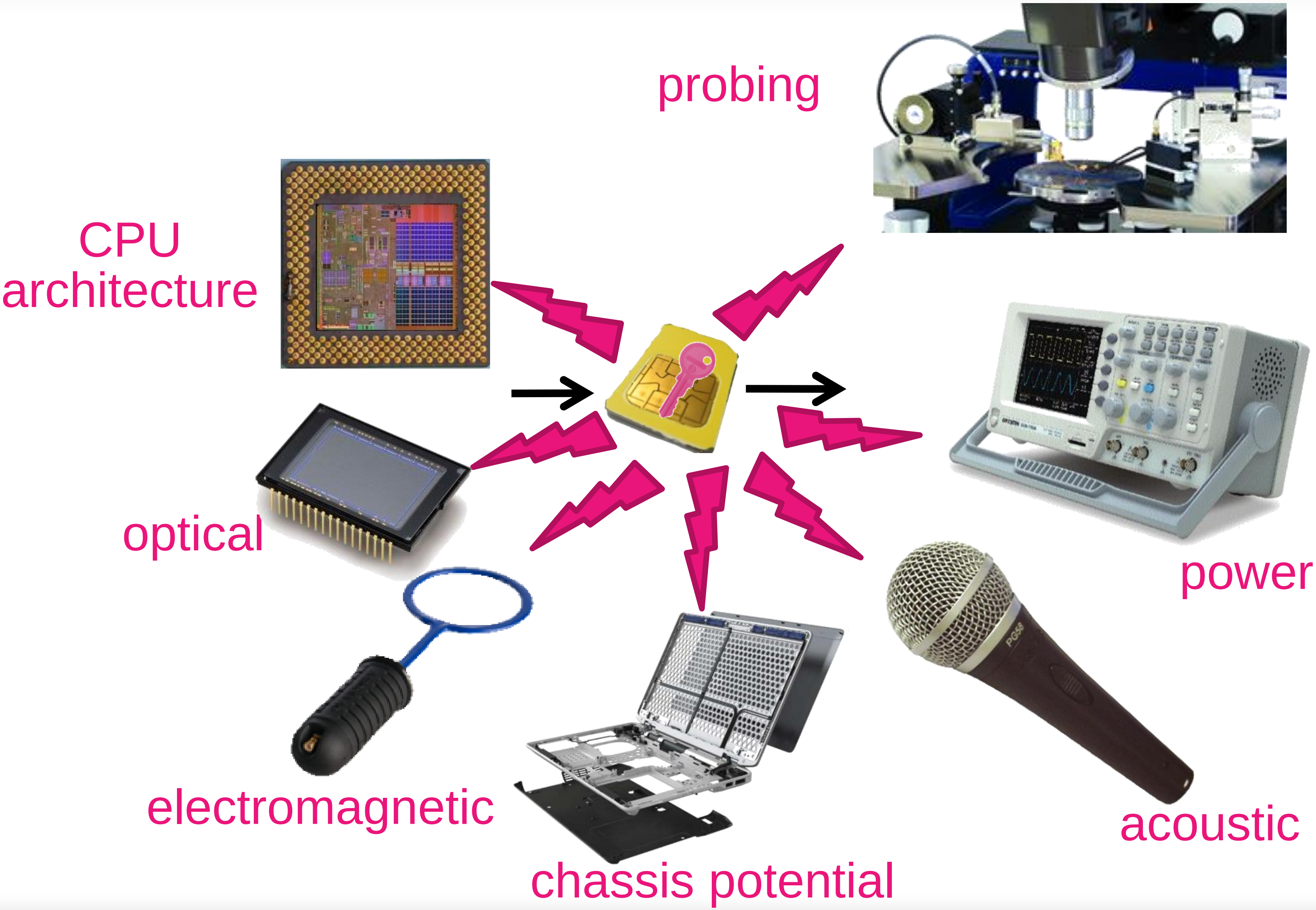
Side channel attacks



Side channel attacks



Side channel attacks



Side channel attack example

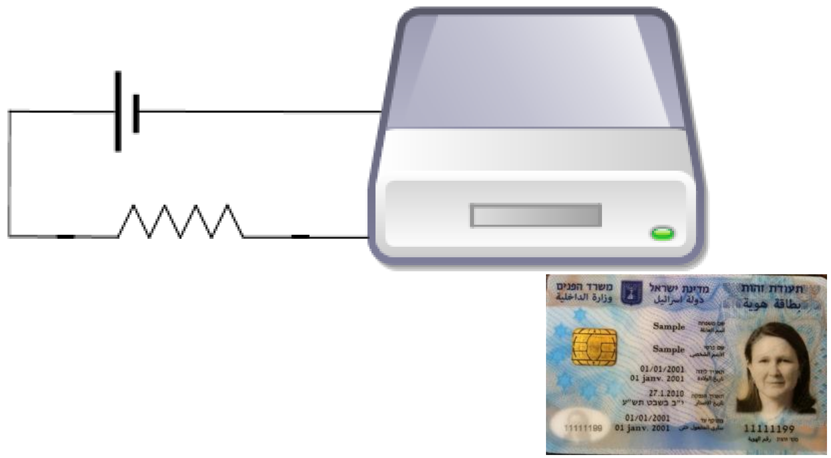
Side channel attack example

100MHz

Side channel attack example

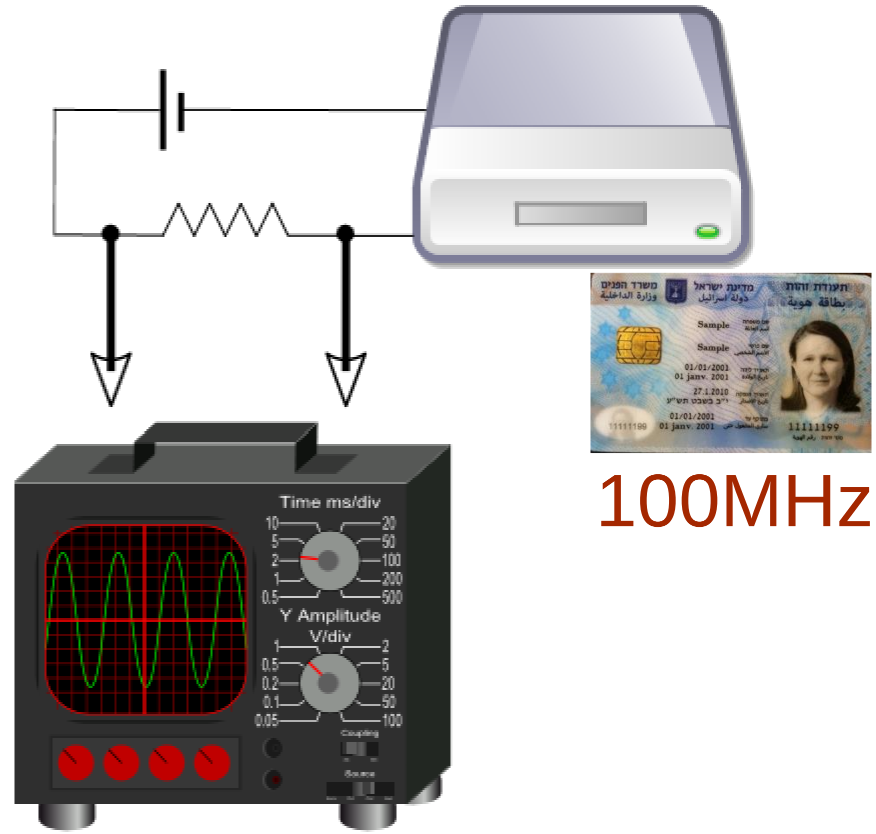
100MHz

Side channel attack example



100MHz

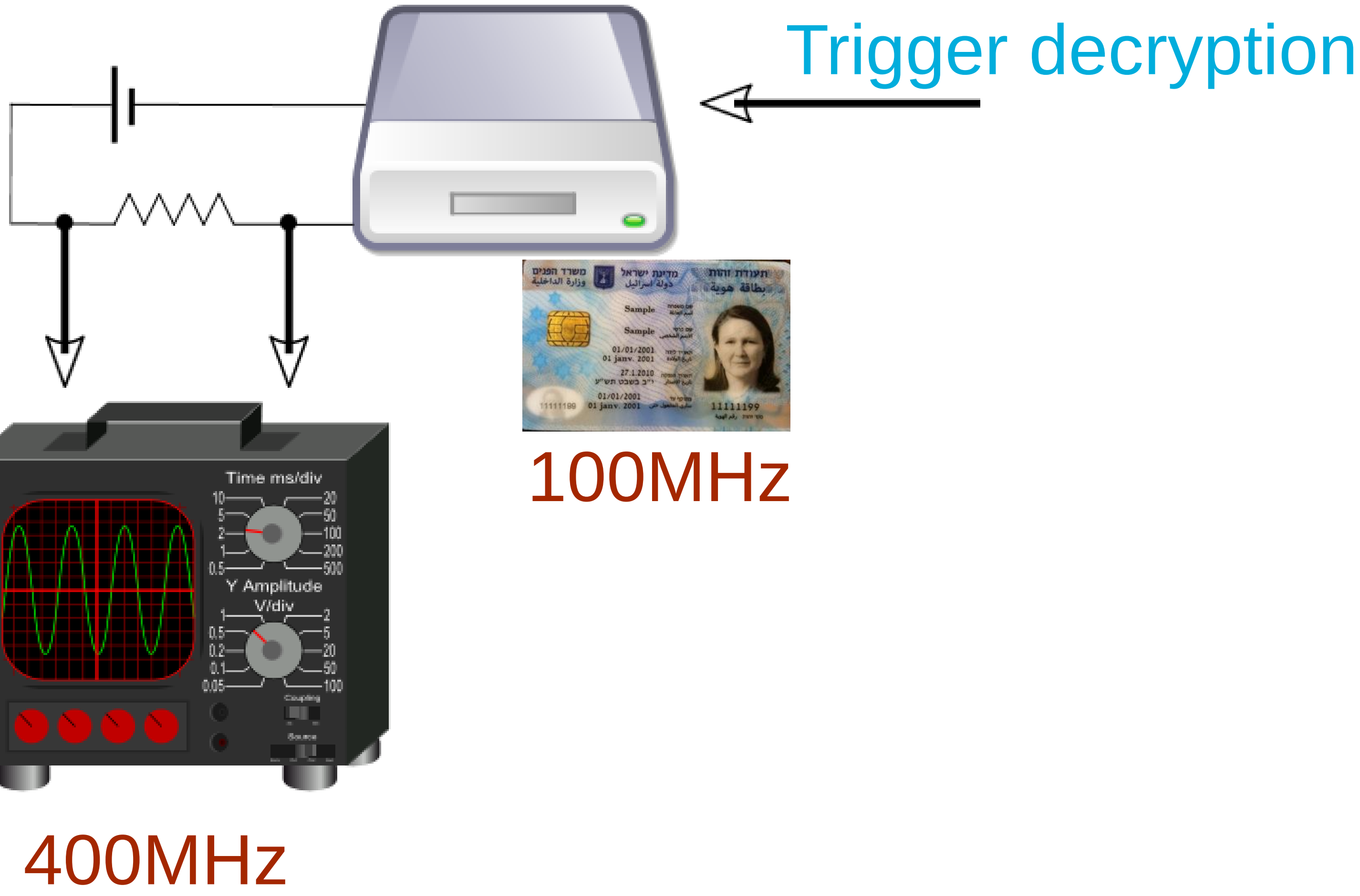
Side channel attack example



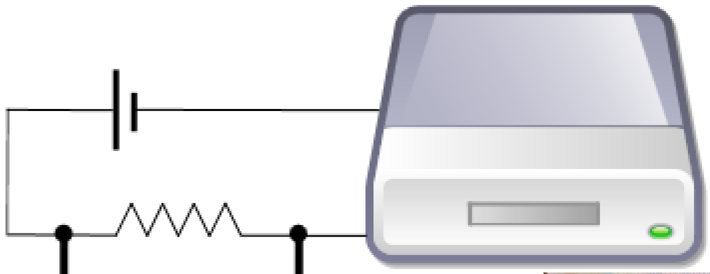
100MHz

400MHz

Side channel attack example

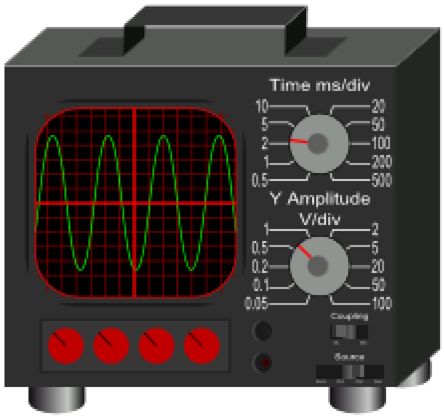


Side channel attack example



Trigger decryption

100MHz



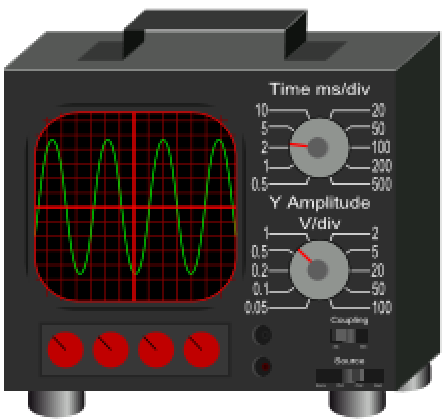
Obtain traces

400MHz

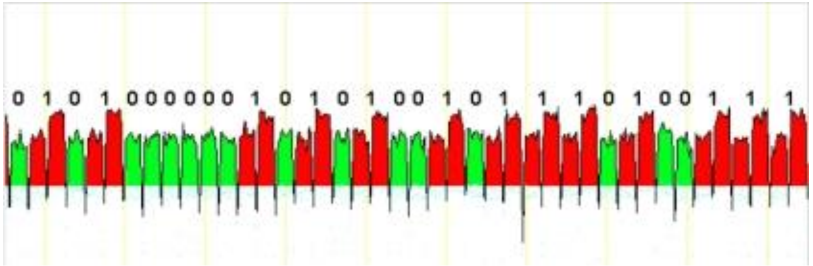
Side channel attack example

Trigger decryption

100MHz



Obtain traces



400MHz

Traditional side channel attacks methodology

1. Grab/borrow/steal device

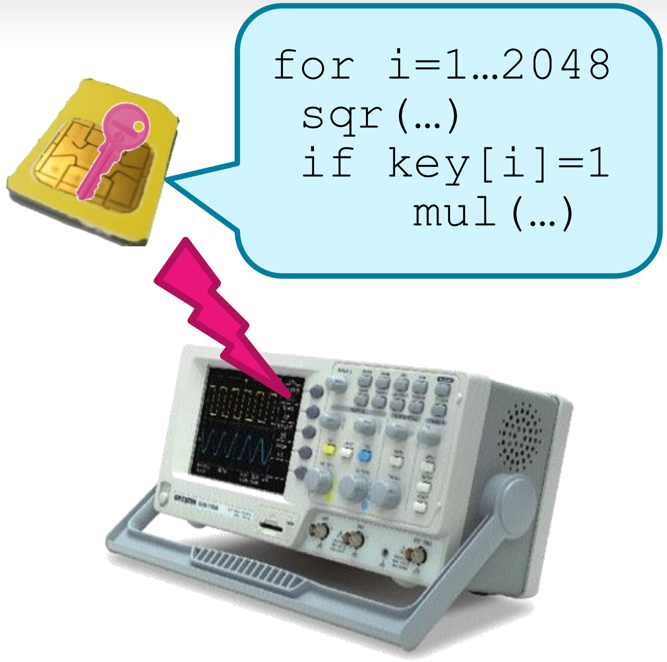
Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction

```
for i=1…2048
 sqr(…)
 if key[i]=1
    mul(…)
```

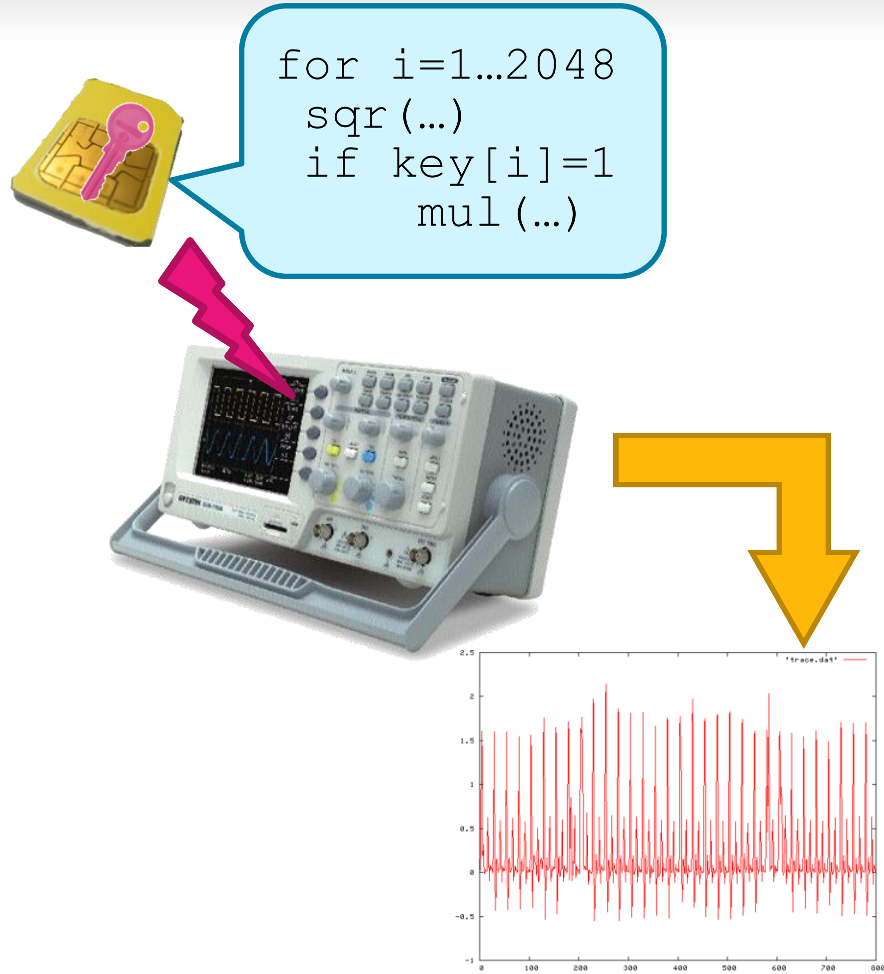
Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)



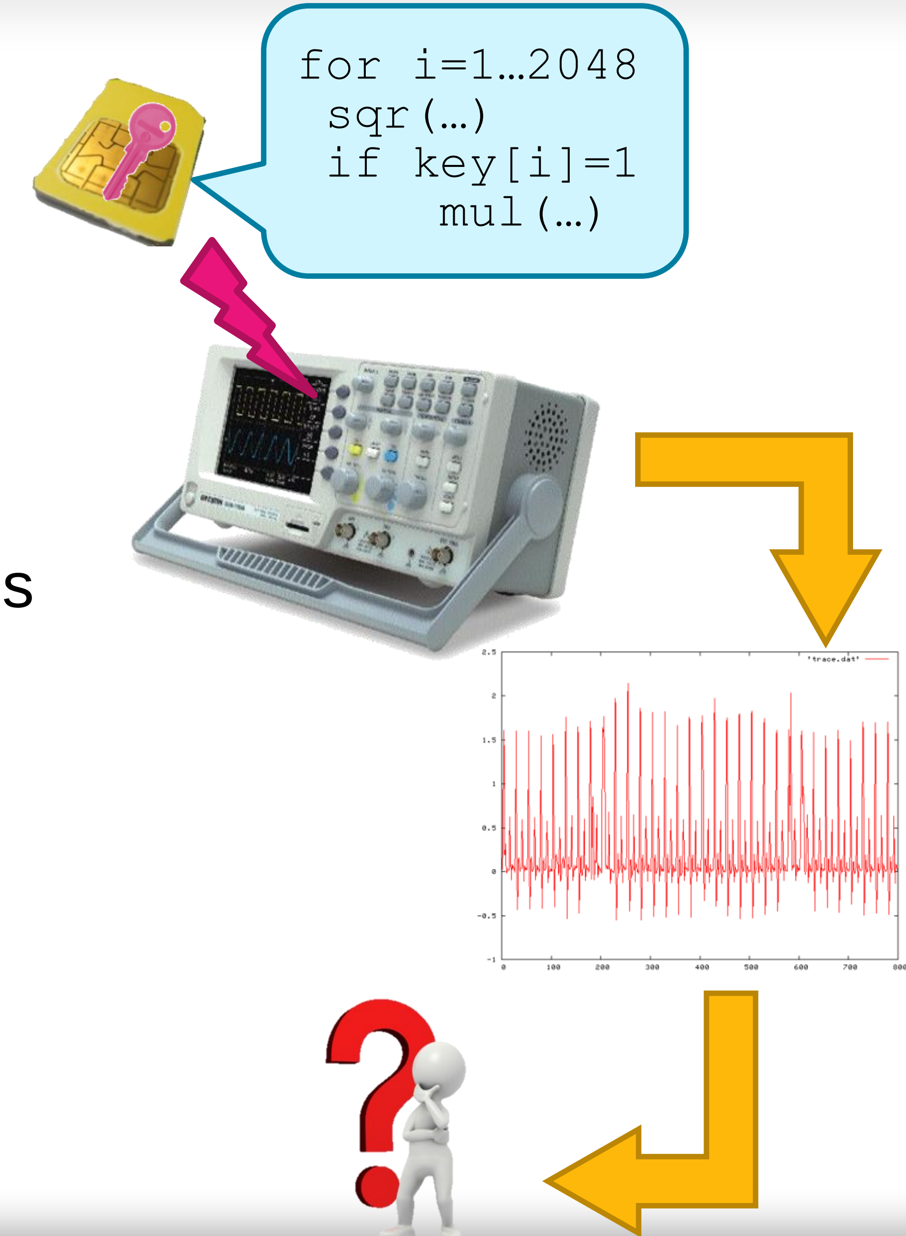
Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)
4. Obtain traces



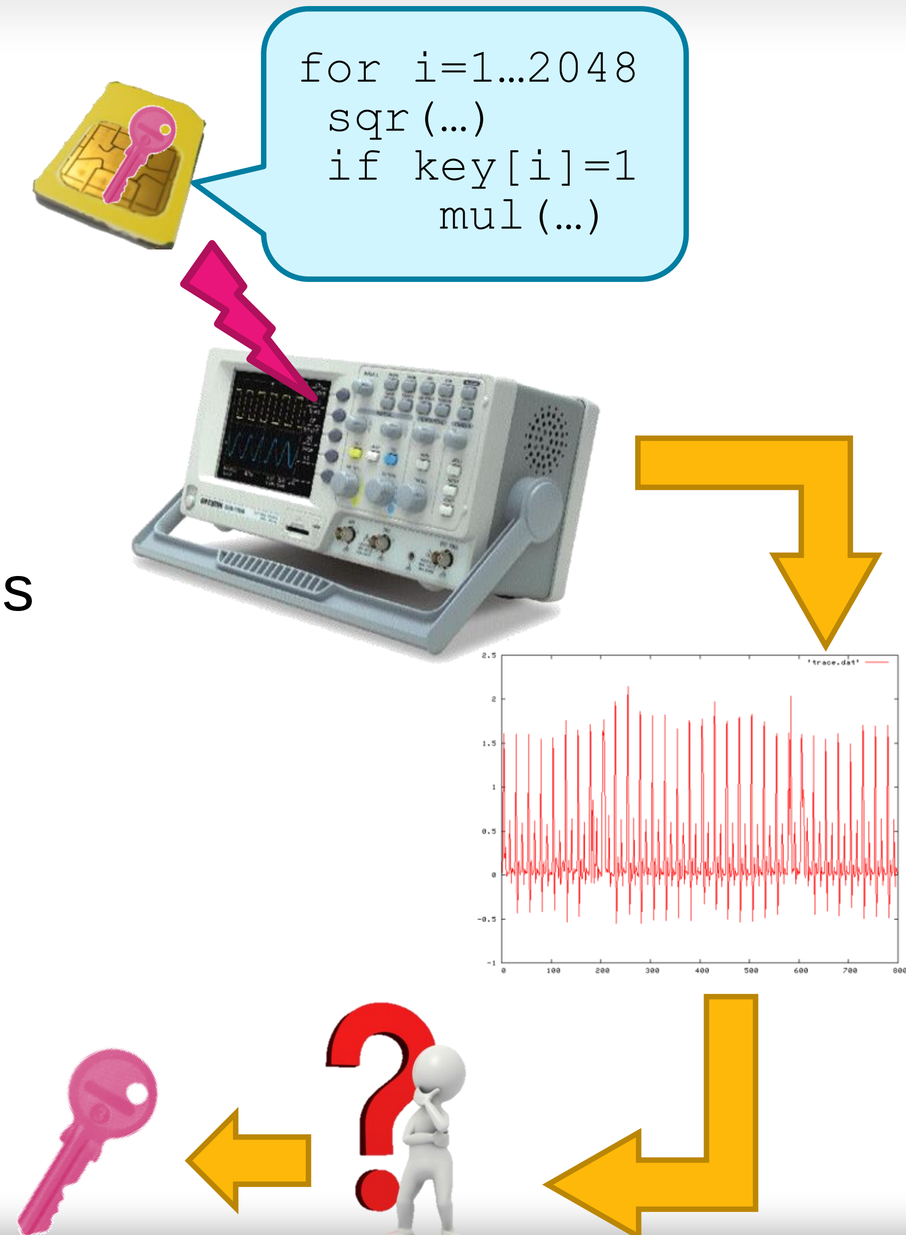
Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)
4. Obtain traces
5. Signal and cryptanalytic analysis



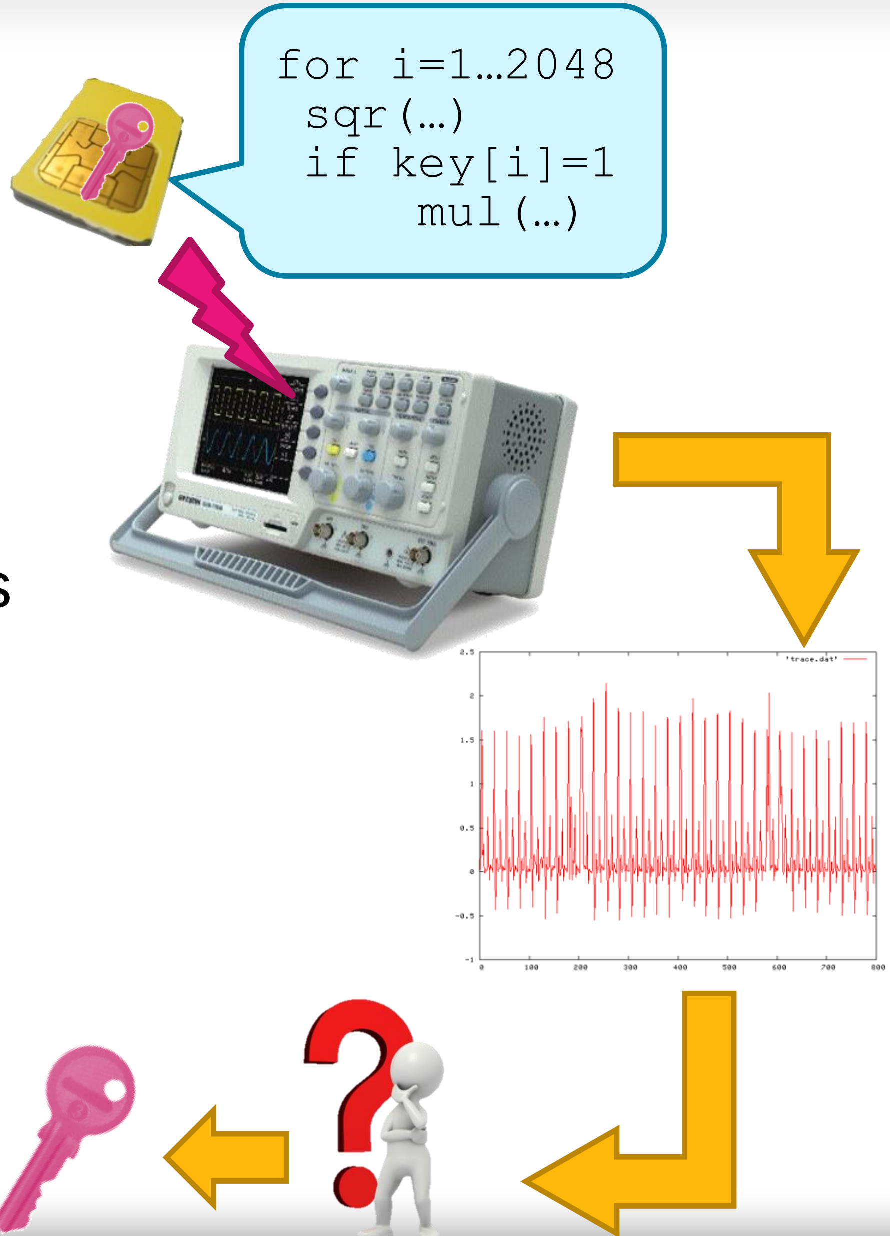
Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)
4. Obtain traces
5. Signal and cryptanalytic analysis
6. Recover key



Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)
4. Obtain traces
5. Signal and cryptanalytic analysis
6. Recover key



Hard for PCs 

Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using
   high-bandwidth equipment
   (> clock rate , PC: >2GHz)
4. Obtain traces
5. Signal and cryptanalytic analysis
6. Recover key

Hard for PCs 

Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)
4. Obtain traces
5. Signal and cryptanalytic analysis
6. Recover key

Not handed out

vs.

Hard for PCs

Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)
4. Obtain traces
5. Signal and cryptanalytic analysis
6. Recover key

Hard for PCs 

Not handed out

 vs. 

Measuring a 2GHz PC requires expansive and bulky equipment (compared to a 100 MHz smart card)

 vs. 

100,000$ 1,000$

Traditional side channel attacks methodology

1. Grab/borrow/steal device
2. Find key-dependent instruction
3. Record emanations using high-bandwidth equipment (> clock rate , PC: >2GHz)
4. Obtain traces
5. Signal and cryptanalytic analysis
6. Recover key

Not handed out

vs.

Measuring a 2GHz PC requires expansive and bulky equipment (compared to a 100 MHz smart card)

100,000$

vs.

1,000$

Complex electronics running complicated software (in parallel)

vs.

New channel:
chassis potential

Ground-potential analysis

- **Attenuating EMI emanations**
  “Unwanted currents or electromagnetic fields?
  Dump them to the circuit ground!”
  (Bypass capacitors, RF shields, …)

Ground-potential analysis

- **Attenuating EMI emanations**
  “Unwanted currents or electromagnetic fields?
  Dump them to the circuit ground!”
  (Bypass capacitors, RF shields, …)

Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

Ground-potential analysis

- **Attenuating EMI emanations**
  “Unwanted currents or electromagnetic fields? Dump them to the circuit ground!”
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its “ground” potential fluctuates relative to the mains earth ground.

Computation

Ground-potential analysis

- **Attenuating EMI emanations**
  “Unwanted currents or electromagnetic fields? Dump them to the circuit ground!”
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its “ground” potential fluctuates relative to the mains earth ground.

Computation

affects currents and EM fields

Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground

Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

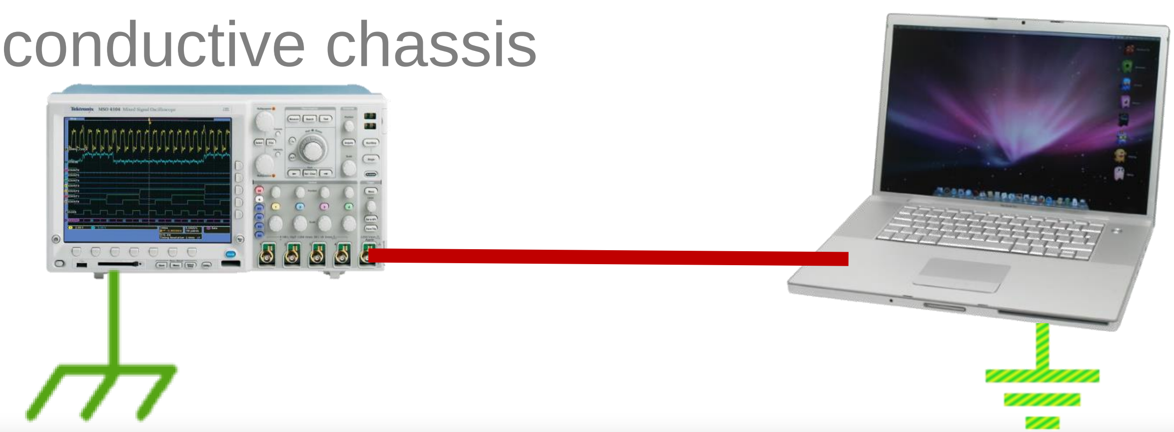
	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis



Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

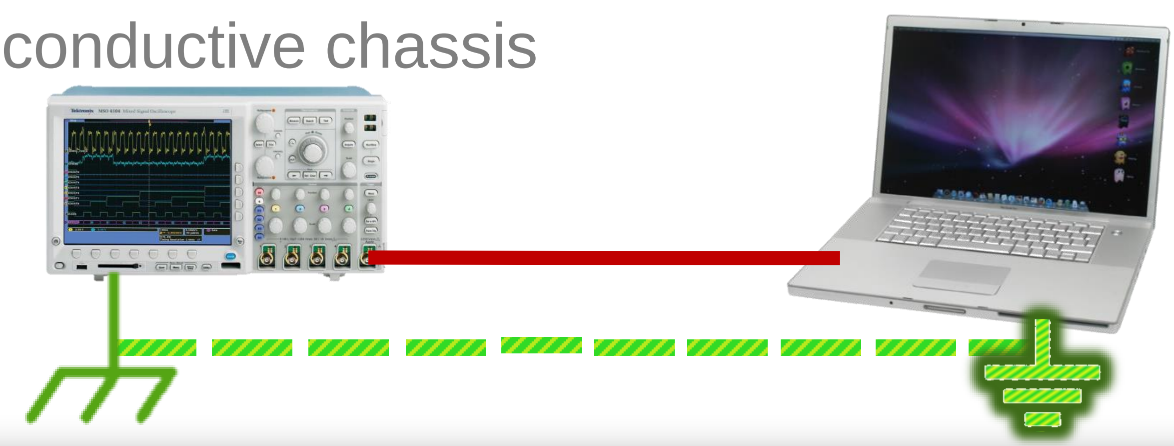
	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

Ground-potential analysis

- **Attenuating EMI emanations**
  “Unwanted currents or electromagnetic fields? Dump them to the circuit ground!”
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its “ground” potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

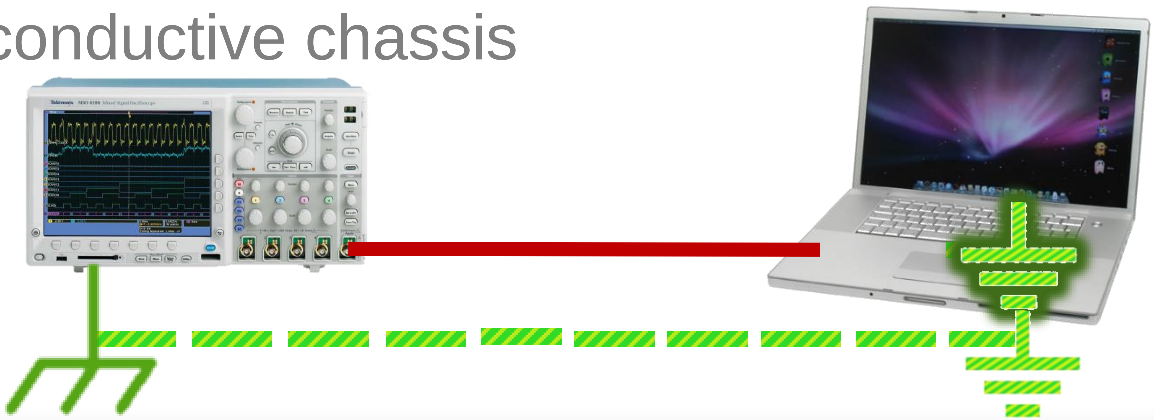


Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

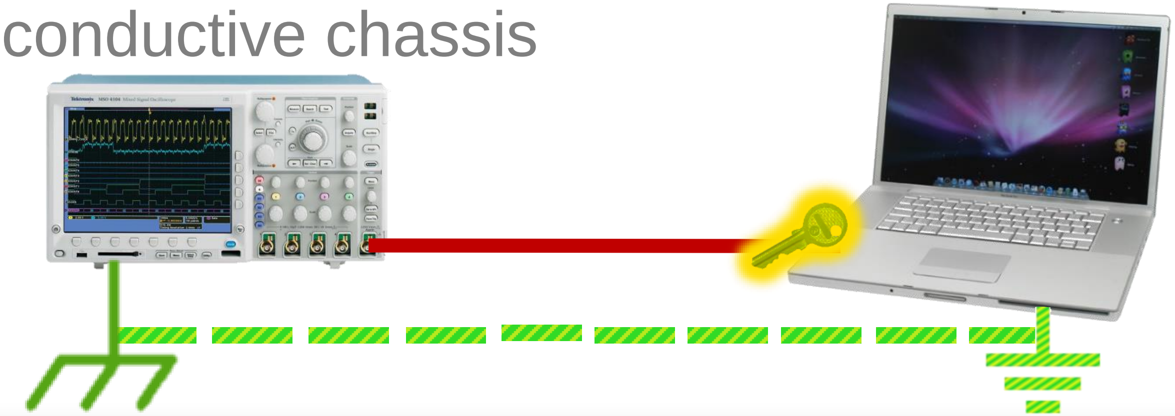


Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

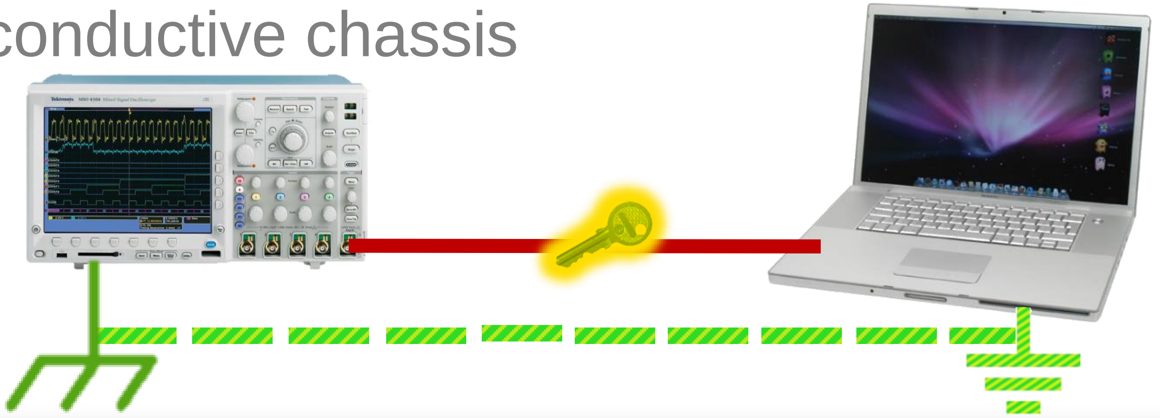


Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

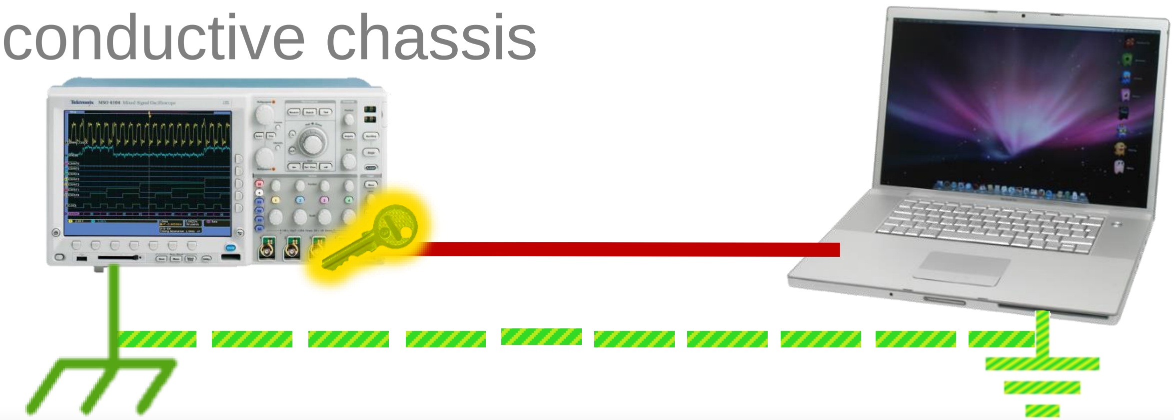


Ground-potential analysis

- **Attenuating EMI emanations**
  “Unwanted currents or electromagnetic fields? Dump them to the circuit ground!”
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its “ground” potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

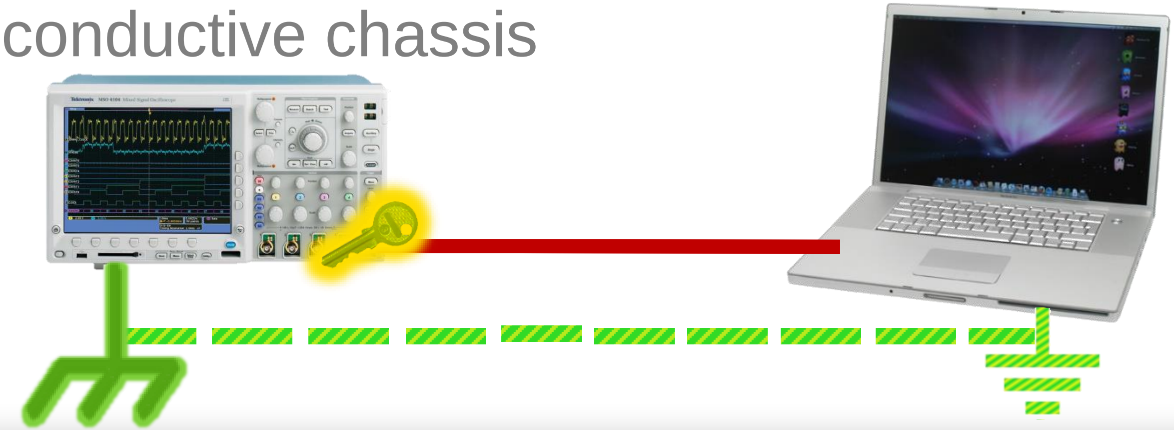


Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

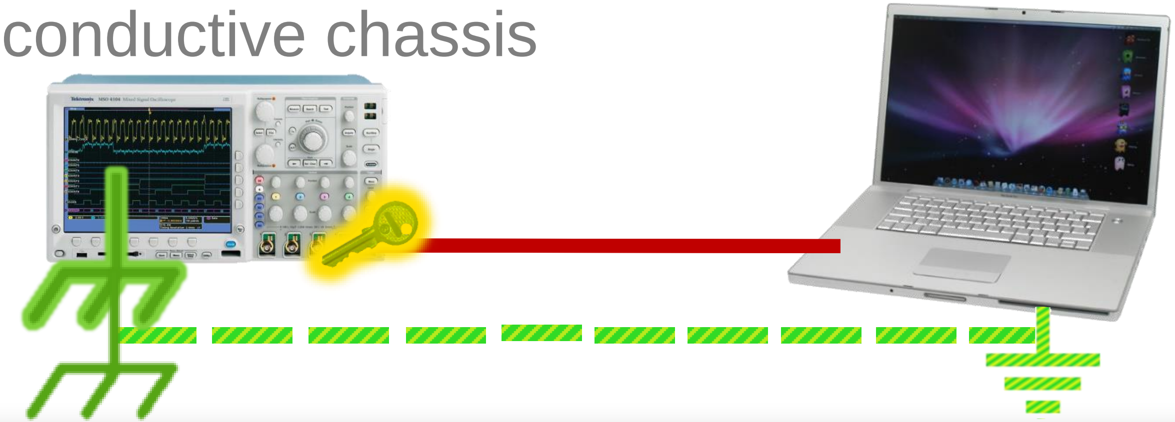


Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

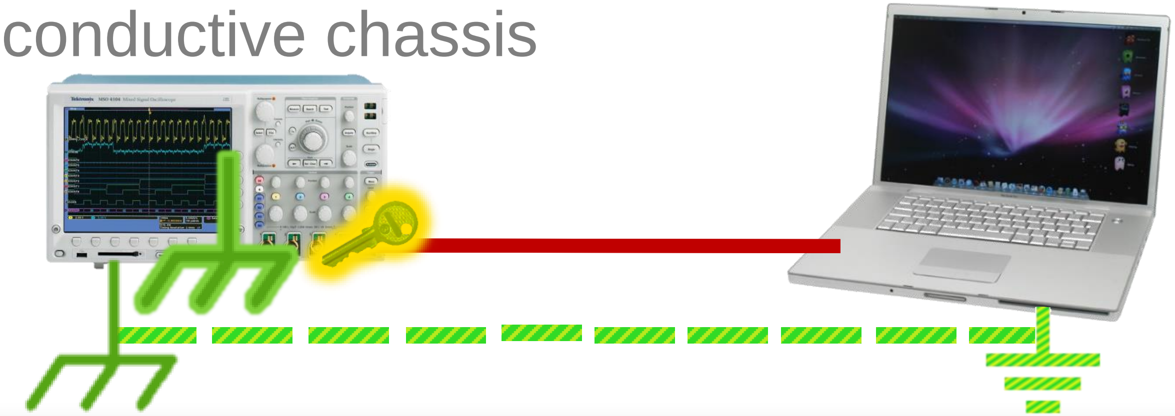


Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

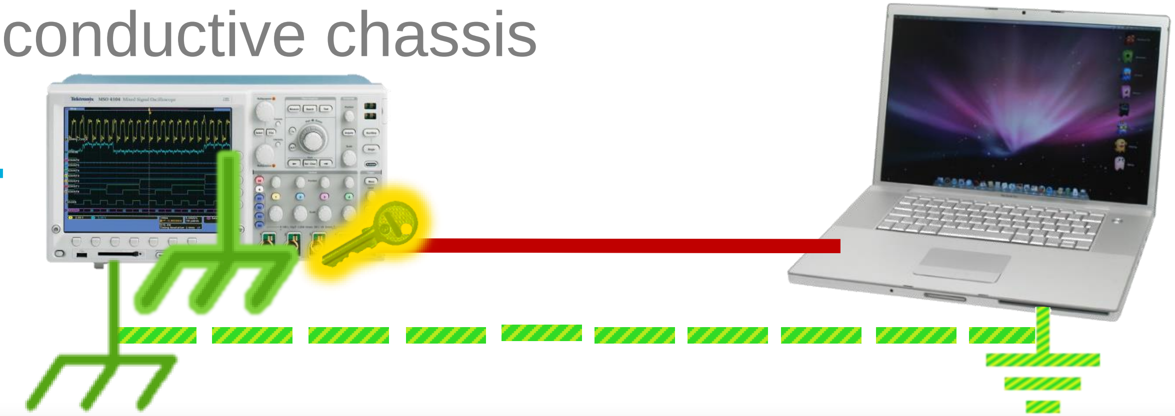


Ground-potential analysis

- **Attenuating EMI emanations**
  “Unwanted currents or electromagnetic fields? Dump them to the circuit ground!”
  (Bypass capacitors, RF shields, …)
- Device is grounded, but its “ground” potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

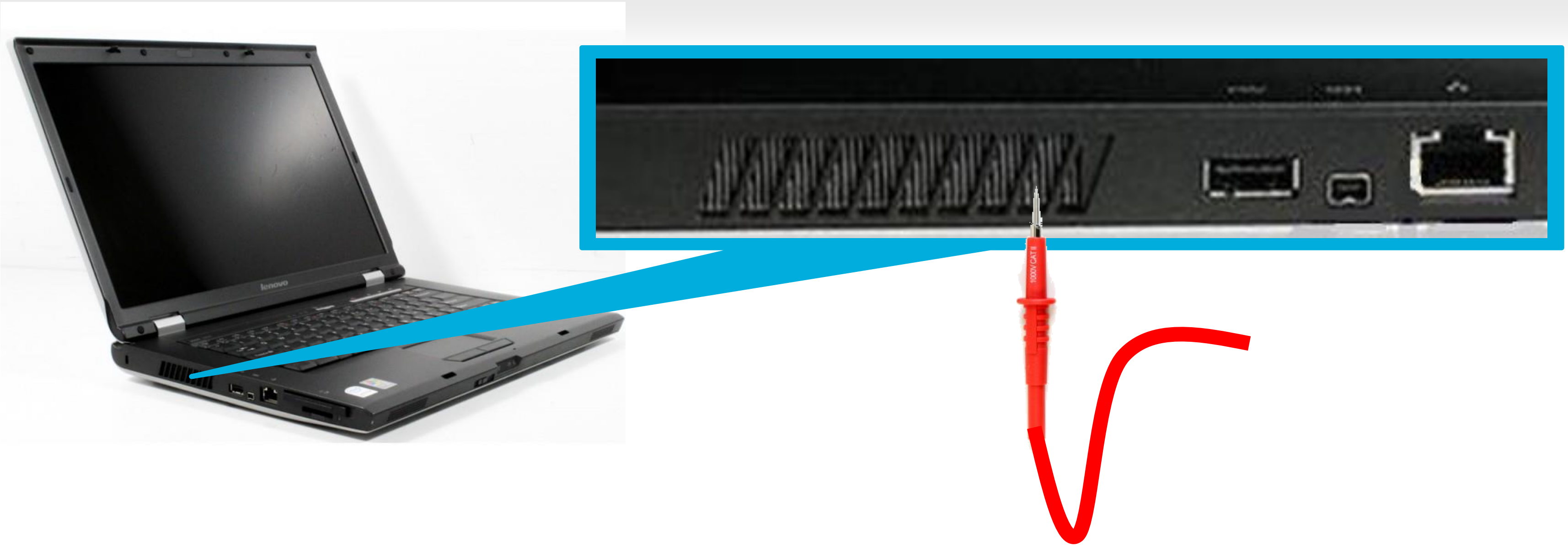
Key = 101011…



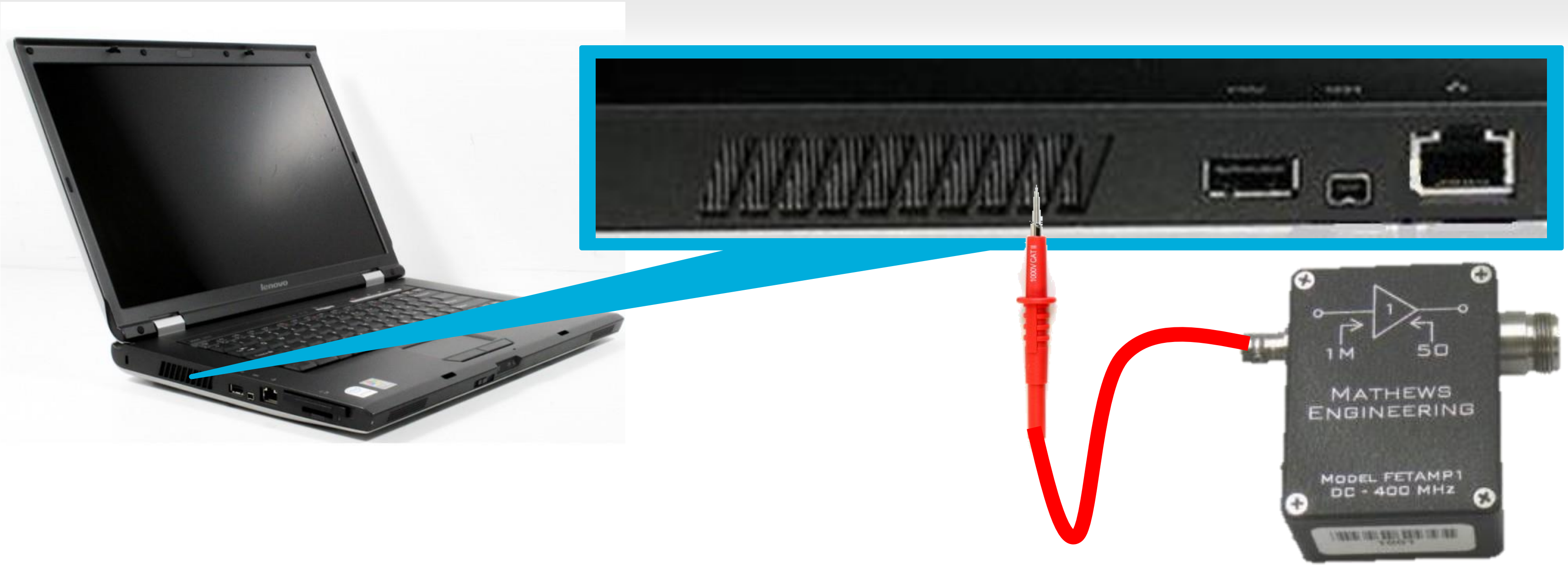
Connecting to the chassis

Connecting to the chassis

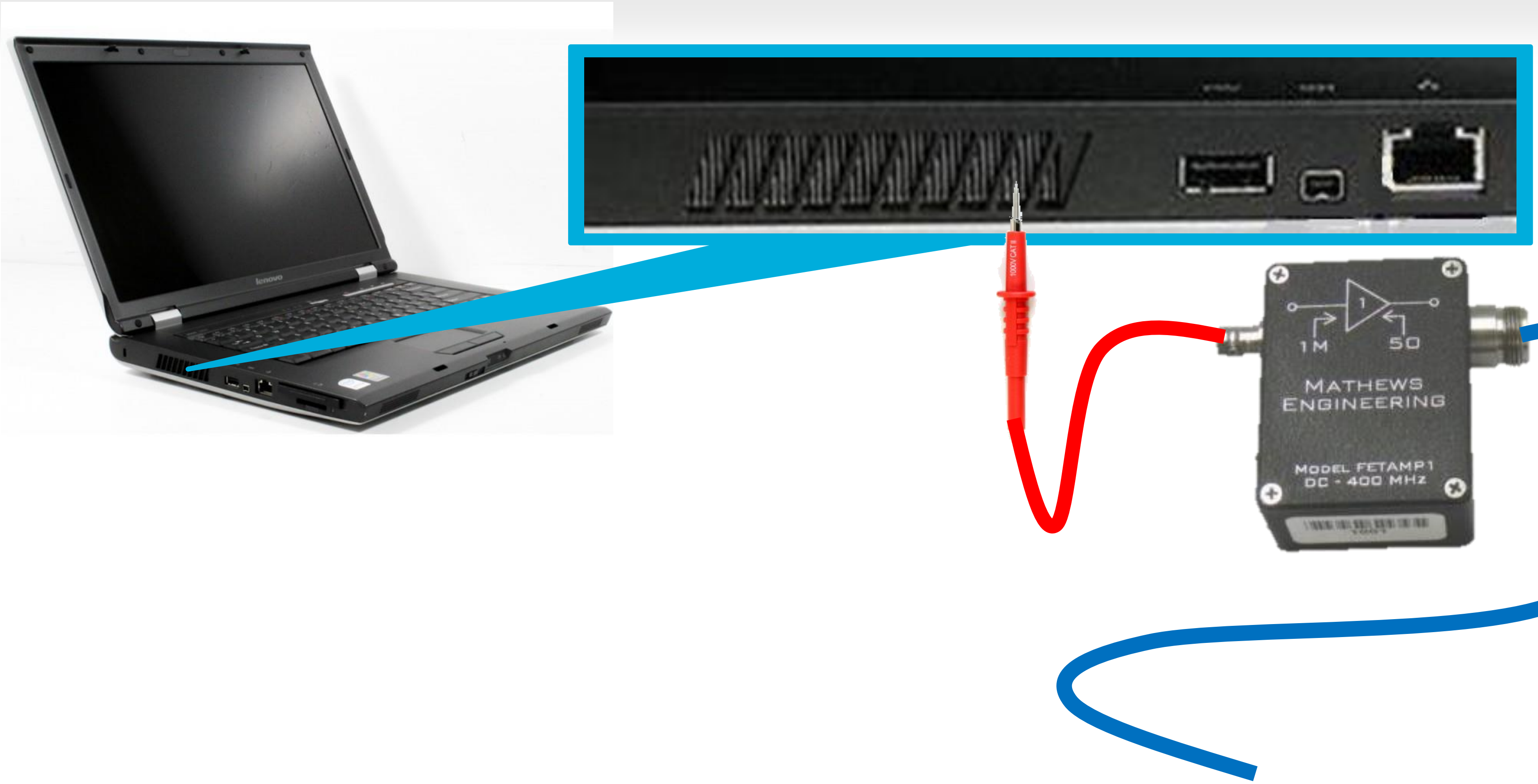
Connecting to the chassis



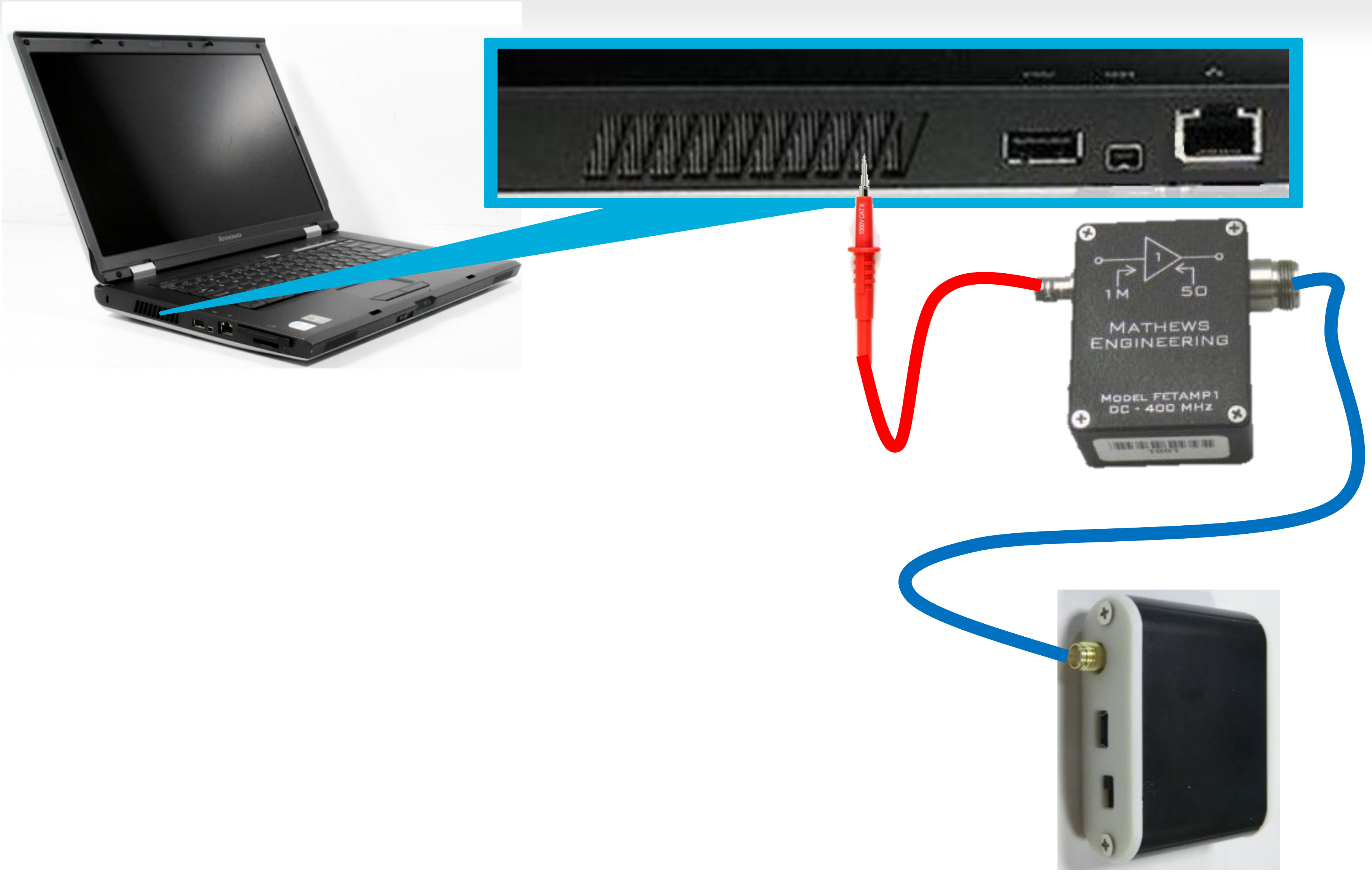
Connecting to the chassis



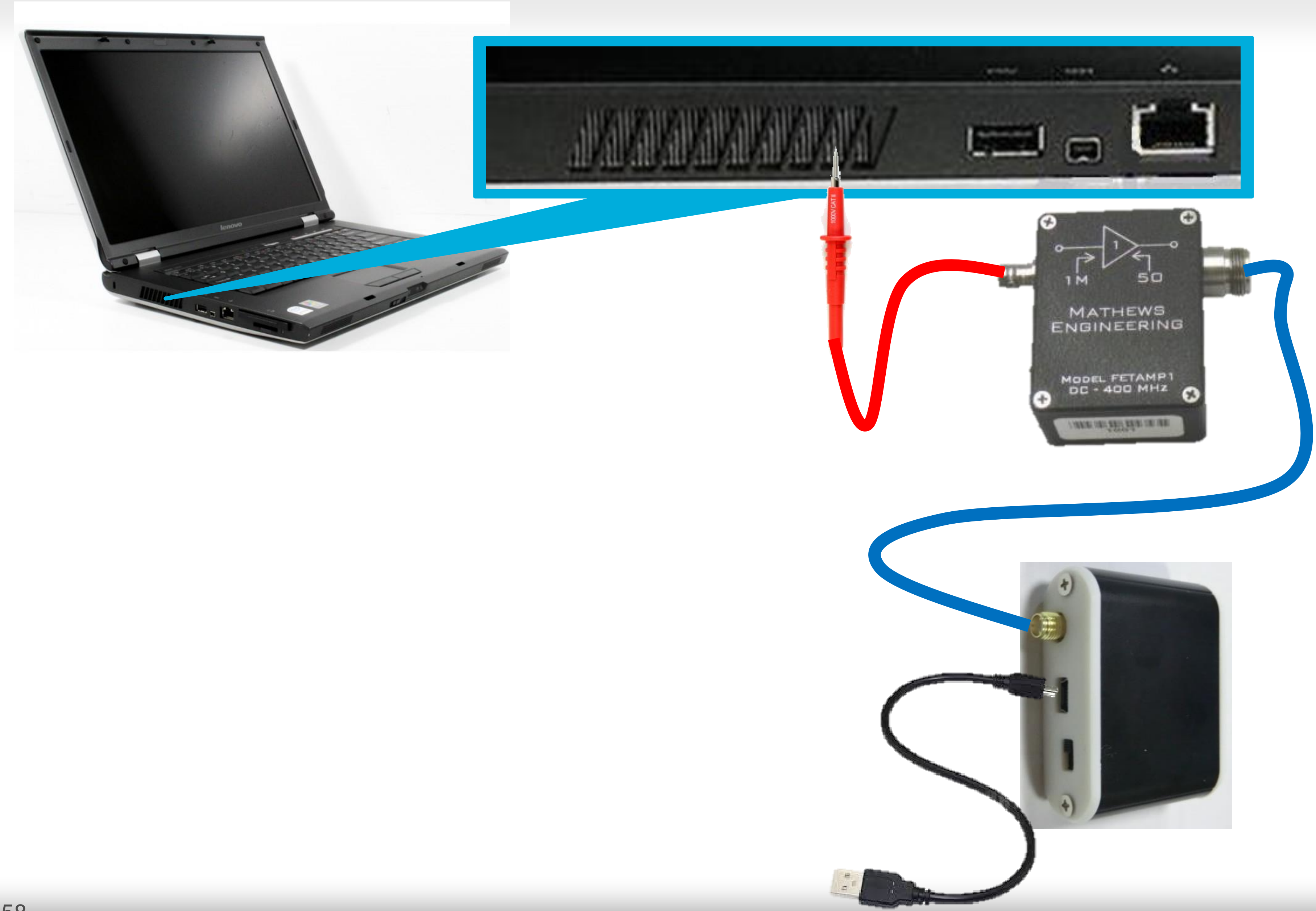
Connecting to the chassis



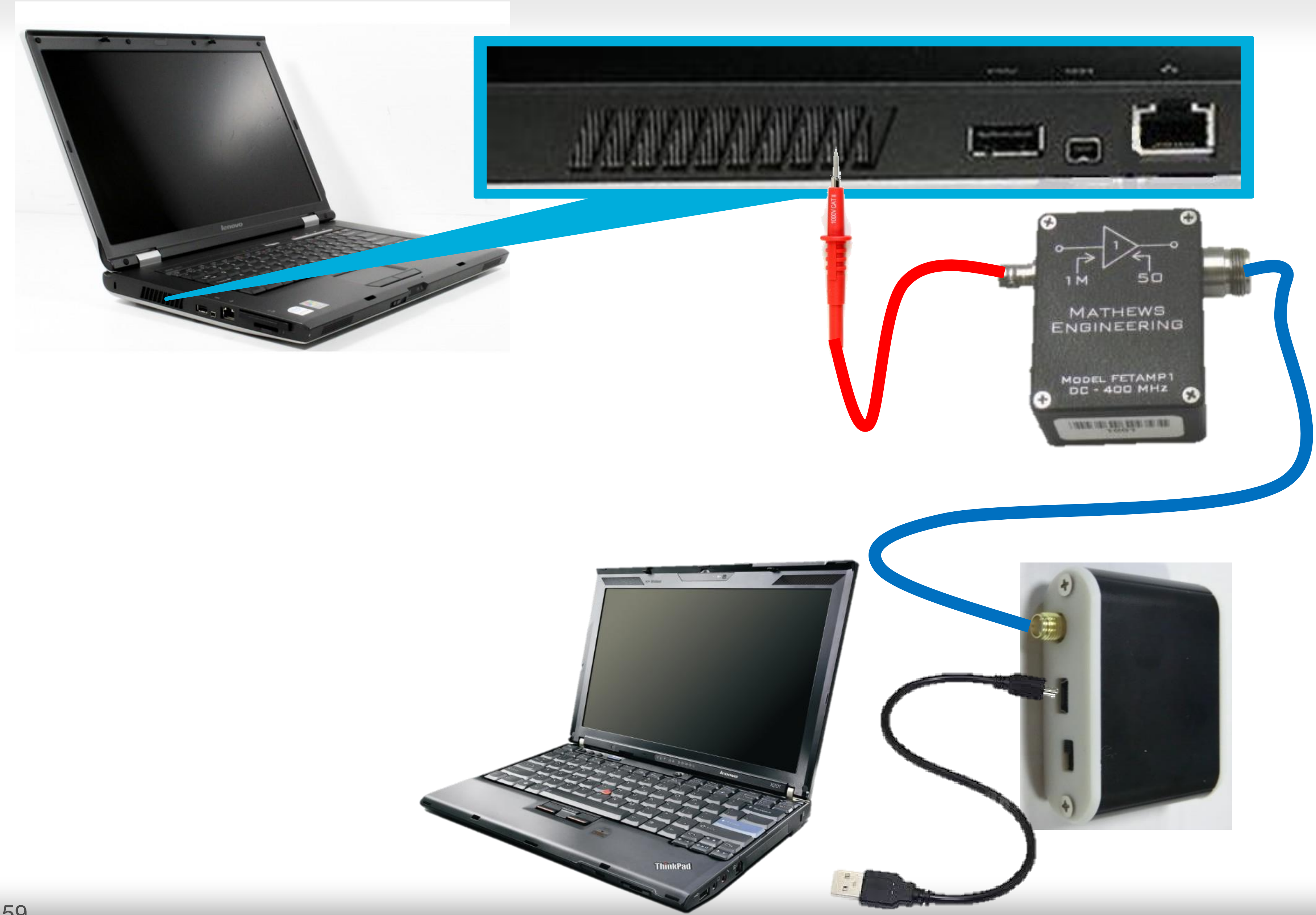
Connecting to the chassis



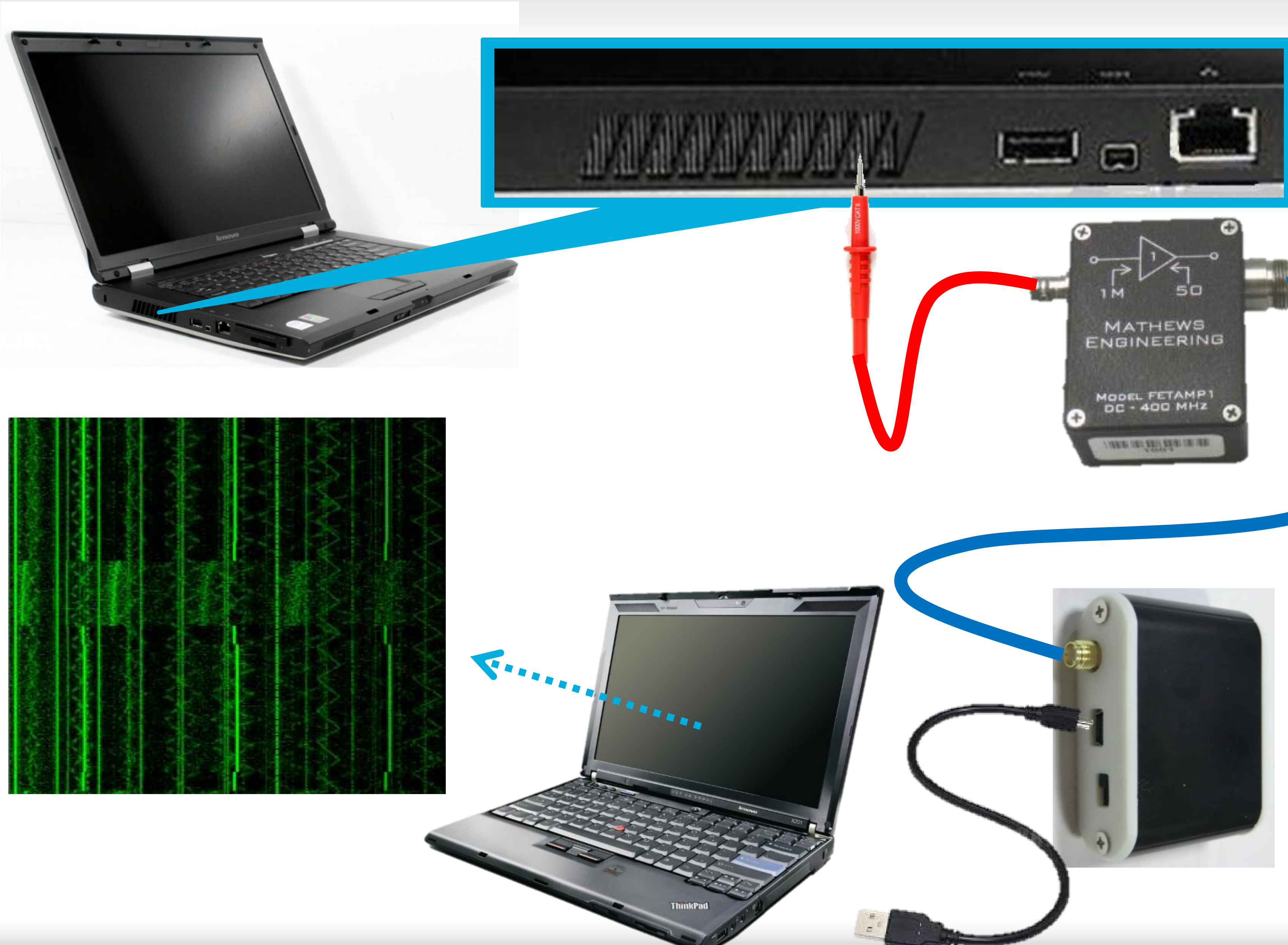
Connecting to the chassis



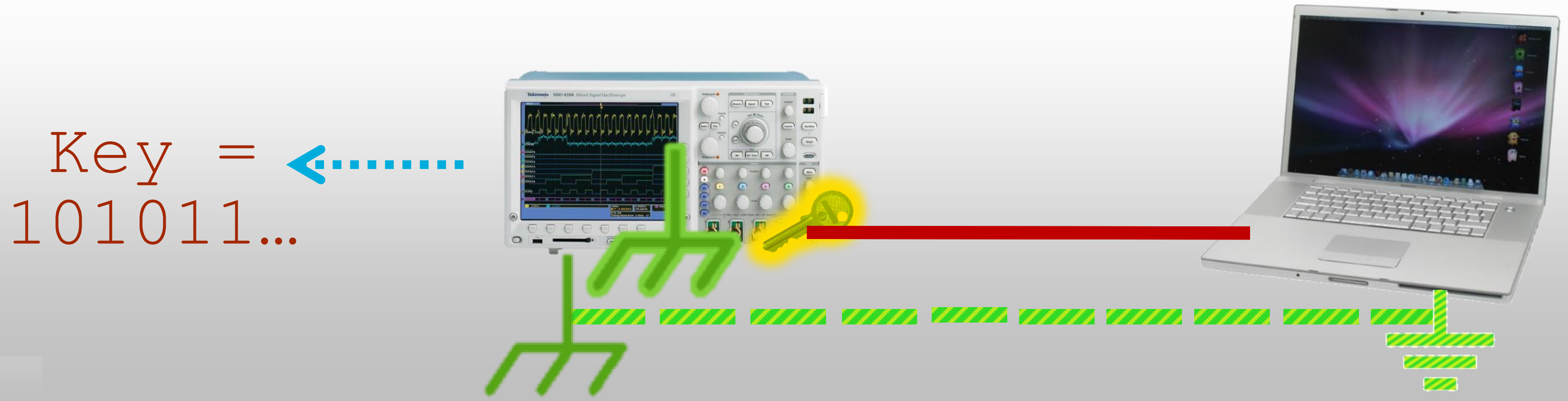
Connecting to the chassis



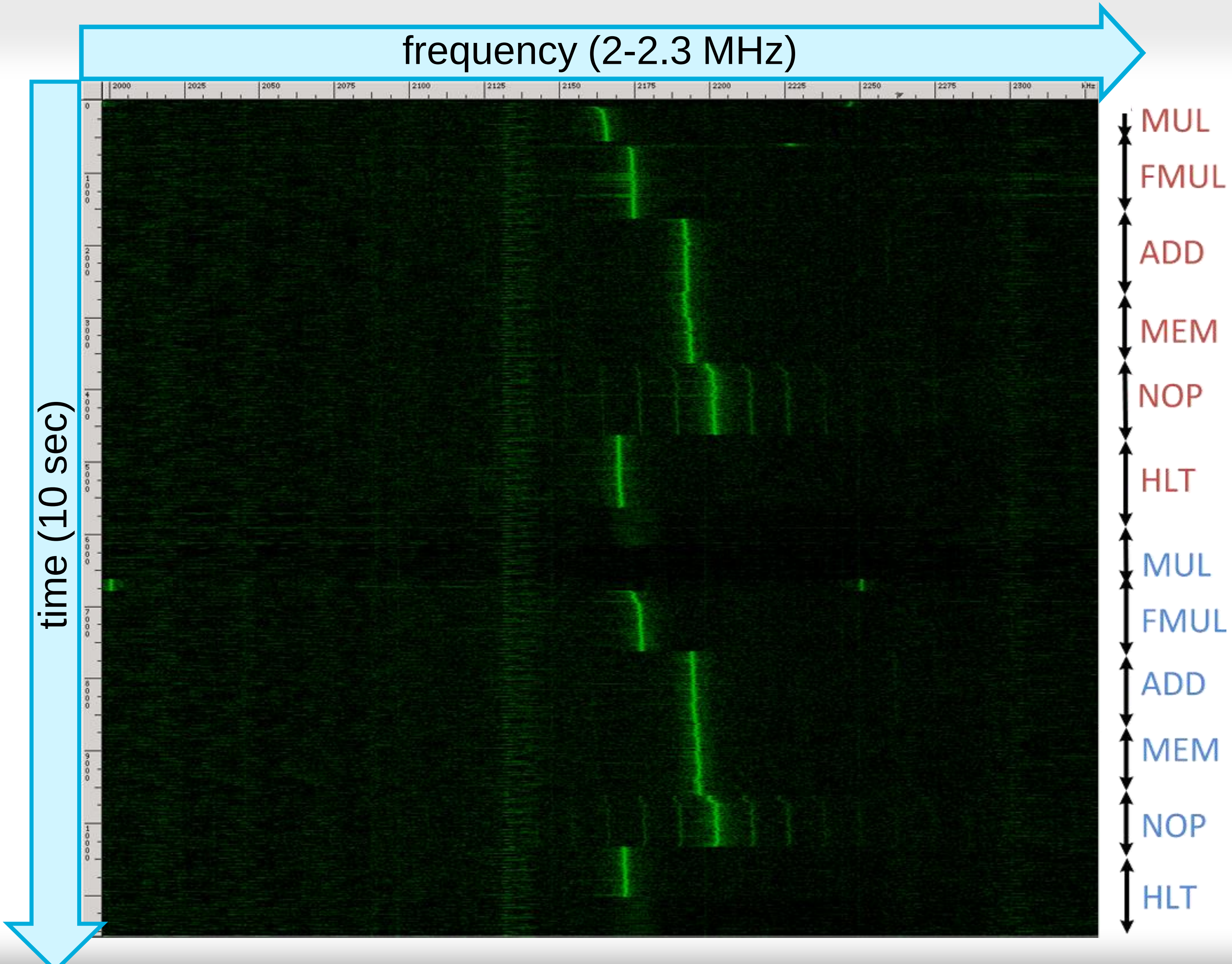
Connecting to the chassis



Demo:
distinguishing instructions



Distinguishing various CPU operations



Our results

- Channels for attacking PCs
  - Ground potential (chassis and others)
  - Power
  - Electromagnetic

Our results

- Channels for attacking PCs
  - Ground potential (chassis and others)
  - Power
  - Electromagnetic

- Exploited via low-bandwidth cryptanalytic attacks
  - Adaptive attack (50 kHz bandwidth) [Genkin Shamir Tromer 14]
  - Non-adaptive attack (1.5 MHz bandwidth)

Our results

- Channels for attacking PCs
  - Ground potential (chassis and others)
  - Power
  - Electromagnetic

- Exploited via low-bandwidth cryptanalytic attacks
  - Adaptive attack (50 kHz bandwidth) [Genkin Shamir Tromer 14]
  - Non-adaptive attack (1.5 MHz bandwidth)

- Common cryptographic software
  - GnuPG 1.4.15 (CVE 2013-4576, CVE-2014-5270)
  - RSA, ElGamal
  - Worked with GnuPG developers to mitigate the attack

GnuPG

Our results

- Channels for attacking PCs
  - Ground potential (chassis and others)
  - Power
  - Electromagnetic

- Exploited via low-bandwidth cryptanalytic attacks
  - Adaptive attack (50 kHz bandwidth) [Genkin Shamir Tromer 14]
  - Non-adaptive attack (1.5 MHz bandwidth)

- Common cryptographic software
  - GnuPG 1.4.15 (CVE 2013-4576, CVE-2014-5270)
  - RSA, ElGamal
  - Worked with GnuPG developers to mitigate the attack

GnuPG

- Applicable to various laptop models

Our results

- Channels for attacking PCs
  - Ground potential (chassis and others)
  - Power
  - Electromagnetic

- Exploited via low-bandwidth cryptanalytic attacks
  - Adaptive attack (50 kHz bandwidth) [Genkin Shamir Tromer 14]
  - Non-adaptive attack (1.5 MHz bandwidth) **today**

- Common cryptographic software
  - GnuPG 1.4.15 (CVE 2013-4576, CVE-2014-5270)
  - RSA, ElGamal
  - Worked with GnuPG developers to mitigate the attack

- Applicable to various laptop models

Low-bandwidth leakage of RSA

Definitions (RSA)

Key setup

- **sk:** random primes p, q, private exponent d
- **pk:** $n = pq$, public exponent e

Encryption

$$c = m^e \; mod \; n$$

Decryption

$$m = c^d \; mod \; n$$

Definitions (RSA)

Key setup

- **sk:** random primes p, q, private exponent d
- **pk:** $n = pq$, public exponent e

Encryption

$$c = m^e \ mod\ n$$

Decryption

$$m = c^d \ mod\ n$$

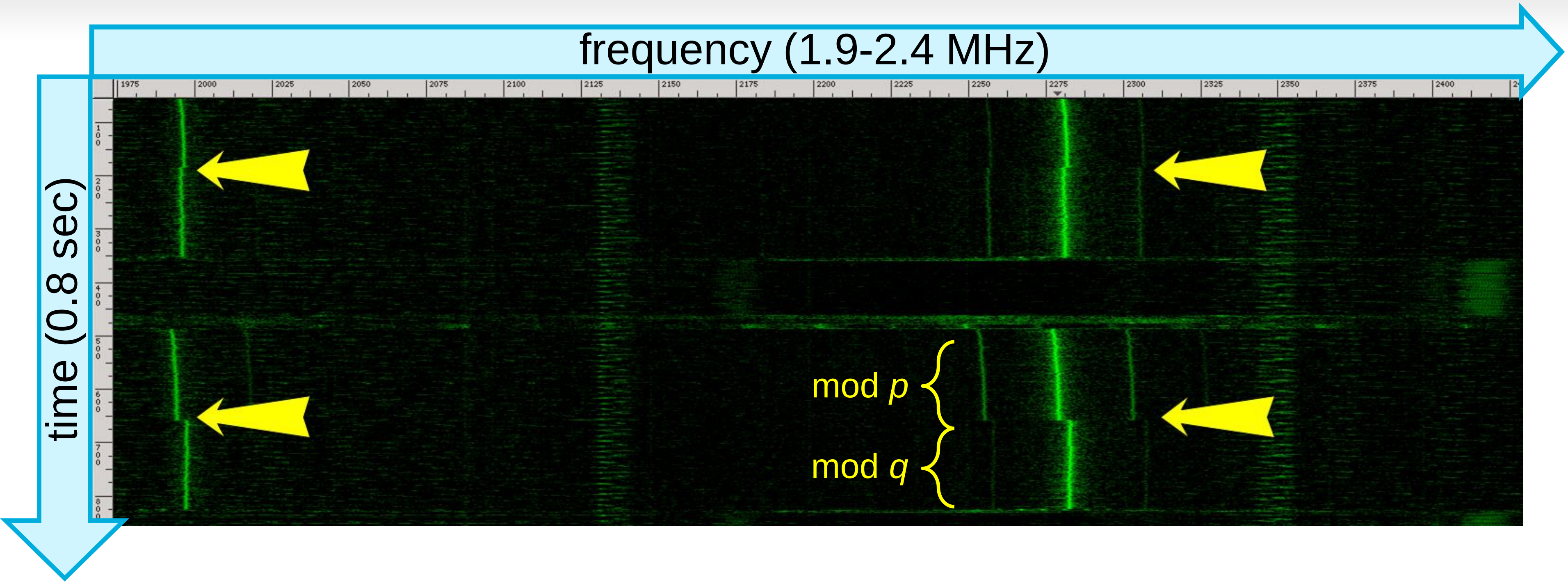
A quicker way used by most implementations

$$m_p = c^{d_p} \ mod\ p$$

$$m_q = c^{d_q} \ mod\ q$$

Obtain m using Chinese Remainder Theorem

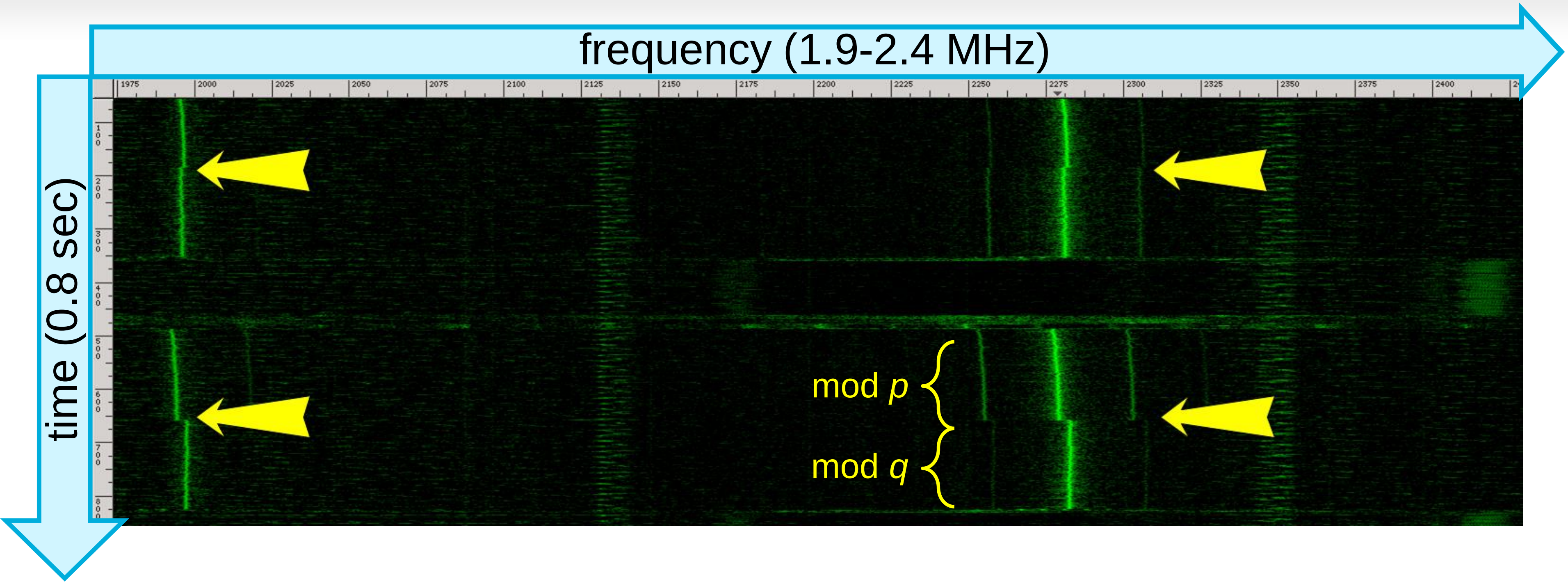
GnuPG RSA key distinguishability



Can distinguish between:

1. Decryptions and other operations

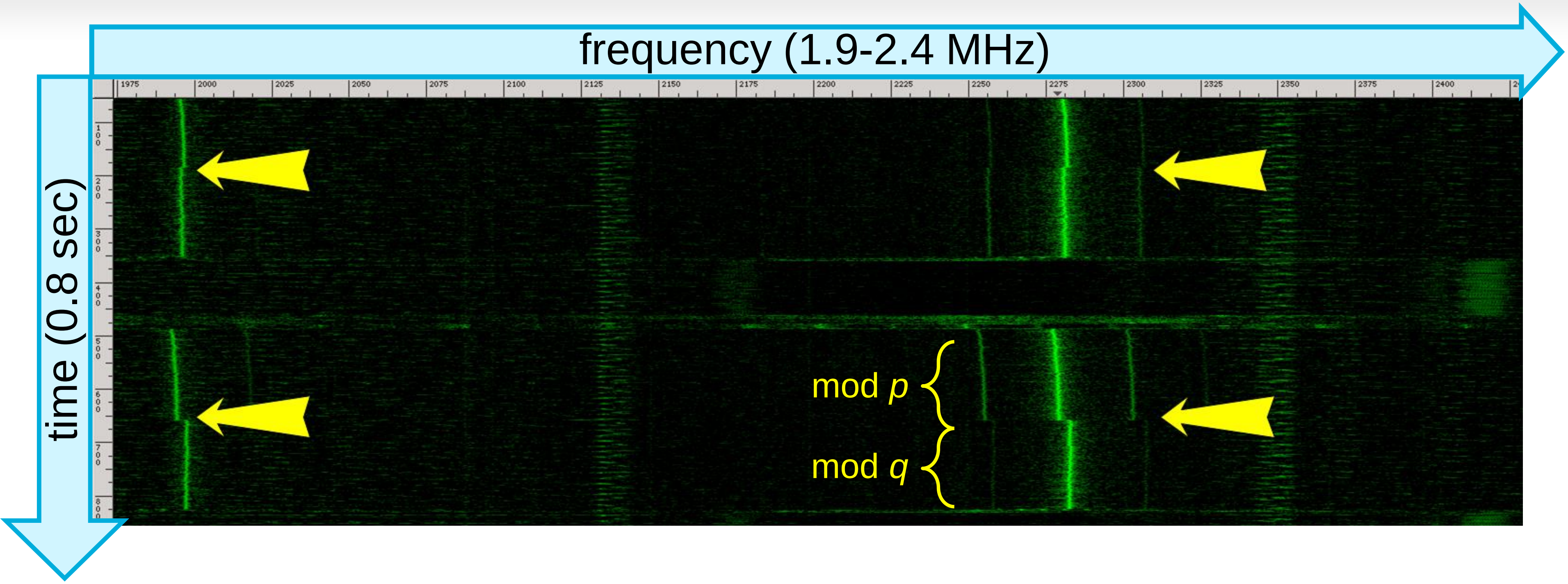
GnuPG RSA key distinguishability



Can distinguish between:

1. Decryptions and other operations
2. Two exponentiations (mod p, mod q)

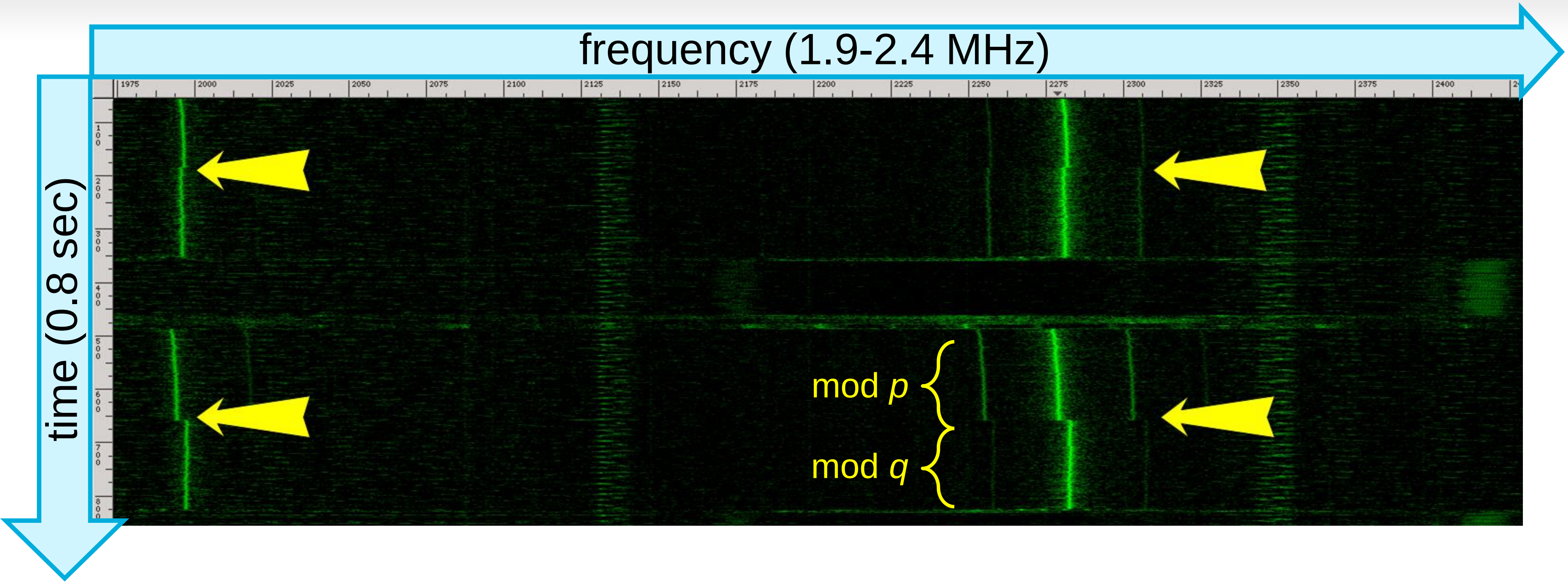
GnuPG RSA key distinguishability



Can distinguish between:

1. Decryptions and other operations
2. Two exponentiations (mod p, mod q)
3. Different keys

GnuPG RSA key distinguishability



Can distinguish between:

1. Decryptions and other operations
2. Two exponentiations (mod p, mod q)
3. Different keys
4. Different primes

Key extraction

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m



}
```

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
}
```

$m = c^{d_n \cdots d_{i+1}} \; mod \; p$

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
}
```

$\mathrm{m} = c^{d_n \cdots d_{i+1}}\ mod\ p$

$\mathrm{m} = c^{d_n \cdots d_{i+1} 0}\ mod\ p$

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
}
```

$m = c^{d_n \cdots d_{i+1}} \, mod \; p$

$m = c^{d_n \cdots d_{i+1} 0} \, mod \; p$

$t = c^{d_n \cdots d_{i+1} 1} \, mod \; p$

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
}
```

$m = c^{d_n \cdots d_{i+1}} \; mod \; p$

$m = c^{d_n \cdots d_{i+1} 0} \; mod \; p$

$t = c^{d_n \cdots d_{i+1} 1} \; mod \; p$

$m = c^{d_n \cdots d_i} \; mod \; p$

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
}
```

$\mathrm{m} = c^{d_n \cdots d_{i+1}} \, mod \; p$

$\mathrm{m} = c^{d_n \cdots d_{i+1} 0} \, mod \; p$

$t = c^{d_n \cdots d_{i+1} 1} \, mod \; p$

$\mathrm{m} = c^{d_n \cdots d_i} \, mod \; p$

Q: Why always compute $t \leftarrow m \cdot c$ then conditionally copy?
A: This is a side channel countermeasure meant to protect d

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
```

$\mathrm{m} = c^{d_n \cdots d_{i+1}} \mathit{mod}\ p$

$\mathrm{m} = c^{d_n \cdots d_{i+1} 0} \mathit{mod}\ p$

$t = c^{d_n \cdots d_{i+1} 1} \mathit{mod}\ p$

$\mathrm{m} = c^{d_n \cdots d_i} \mathit{mod}\ p$

Q: Why always compute $t \leftarrow m \cdot c$ then conditionally copy?
A: This is a side channel countermeasure meant to protect d


```
}
```

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
}
```

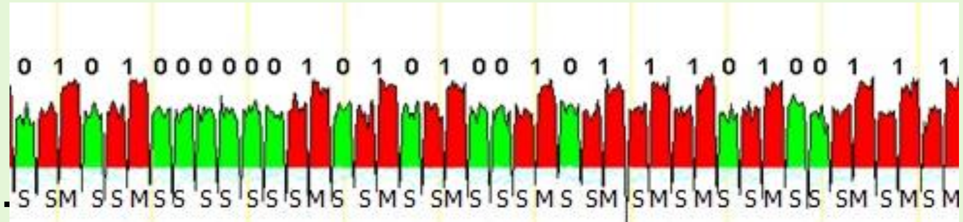
$\mathrm{m} = c^{d_n \cdots d_{i+1}} \, mod\, p$

$\mathrm{m} = c^{d_n \cdots d_{i+1} 0} \, mod\, p$

$t = c^{d_n \cdots d_{i+1} 1} \, mod\, p$

$\mathrm{m} = c^{d_n \cdots d_i} \, mod\, p$

Q: Why always compute $t \leftarrow m \cdot c$ then conditionally copy?
A: This is a side channel countermeasure meant to protect d



no key dependent operation to measure

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m

}
```

m depends on both d_i and c

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m


}
```

m depends on both d_i and c

m is used in next iteration of the main loop

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m

}
```

m depends on both d_i and c

m is used in next iteration of the main loop

craft c to affect m in the next loop iteration, based on d_i

measure changes inside squaring operation and obtain d_i

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m
}
```

m depends on both d_i and c

m is used in next iteration of the main loop

craft c to affect m in the next loop iteration, based on d_i

measure changes inside squaring operation and obtain d_i

can only see drastic changes inside squaring operation

Amplifying the key dependency

- **Difficulties when attacking RSA**
  - **2GHz CPU speed vs. 1.5MHz measurements**
  - **Cannot rely on a single key-dependent instruction**

Amplifying the key dependency

- **Difficulties when attacking RSA**
  - **2GHz CPU speed vs. 1.5MHz measurements**
  - **Cannot rely on a single key-dependent instruction**

- **Idea: <u>leakage self-amplification</u>** [Genkin Shamir Tromer 2014]
  abuse algorithm's own code to amplify its own leakage!

Amplifying the key dependency

- **Difficulties when attacking RSA**
  - **2GHz CPU speed vs. 1.5MHz measurements**
  - **Cannot rely on a single key-dependent instruction**

- **Idea: <u>leakage self-amplification</u>** [Genkin Shamir Tromer 2014]
  abuse algorithm's own code to amplify its own leakage!
  - **Craft suitable cipher-text to affect the inner-most loop**

Amplifying the key dependency

- **Difficulties when attacking RSA**
  - **2GHz CPU speed vs. 1.5MHz measurements**
  - **Cannot rely on a single key-dependent instruction**

- **Idea: <u>leakage self-amplification</u>** [Genkin Shamir Tromer 2014]

  abuse algorithm's own code to amplify its own leakage!
  - **Craft suitable cipher-text to affect the inner-most loop**
  - **Small differences in repeated inner-most loops cause a big overall difference in code behavior**

Amplifying the key dependency

- **Difficulties when attacking RSA**
  - **2GHz CPU speed vs. 1.5MHz measurements**
  - **Cannot rely on a single key-dependent instruction**

- **Idea: <u>leakage self-amplification</u>** [Genkin Shamir Tromer 2014]
  abuse algorithm's own code to amplify its own leakage!
  - **Craft suitable cipher-text to affect the inner-most loop**
  - **Small differences in repeated inner-most loops cause a big overall difference in code behavior**
  - **Measure low-bandwidth leakage**

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c mod p //always mult
  if d[i]=1 then
    m=t
 return m



}
```

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c   d p //always mult
  if d[i]=1
    m=t
 return r

}
```

```
karatsuba_sqr( m ){
 …
 basic_sqr( x )
 …

}
```

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c    d p //always mult
  if d[i]=1
    m=t
 return m
}
```

```
karatsuba_sqr( m ){
 ...
 basic_sqr( x )
 ...
}
```

```
basic_sqr( x ){
...


}
```

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c   d p //always mult
  if d[i]=1
    m=t
 return m
}
```

```
karatsuba_sqr( m ){
 …
 basic_sqr( x )
 …
}
```

```
basic_sqr( x ){
…
 if( x[j]==0)
  y = 0
 else
  y = x[j]*x
}
```

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c    d p //always mult
  if d[i]=1
    m=t
 return r

}
```

```
karatsuba_sqr( m ){
 …
 basic_sqr( x )
 …
}
```

```
basic_sqr( x ){
…
 if( x[j]==0)
  y = 0
 else
  y = x[j]*x
}
```

x7

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c mod p //always mult
  if d[i]=1
    m=t
 return m
}
```

```
karatsuba_sqr( m ){
 …
 basic_sqr( x )
 …
}
```

x27

```
basic_sqr( x ){
…
 if( x[j]==0)
  y = 0
 else
  y = x[j]*x
}
```

x7

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m^2 mod p
  t = m*c mod d p //always mult
  if d[i]=1
    m=t
 return m
}
```

```
karatsuba_sqr( m )
  …
  basic_sqr( x )
  …
}
```

```
basic_sqr( x ){
…
}
```

x27

```
if( x[j]==0)
 y = 0
else
 y = x[j]*x
```

x7

repeated 189 times per bit of d

~0.2ms of measurement per bit of d

GnuPG modular exponentiation

```
modular_exponentiation(c,d,p){
 m=1
 for i=1 to n do
  m = m² mod p
  t = m*c    d p //always mult
  if d[i]=1
    m=t
 return m
}
```

```
karatsuba_sqr( m )
 ...
 basic_sqr( x )
 ...
}
```

x27

```
basic_sqr( x ){
...
 if( x[j]==0)
  y = 0
 else
  y = x[j]*x
}
```

x7

repeated 189 times per bit of d

~0.2ms of measurement per bit of d

craft c such that
$d[i] = 1 \rightarrow x[j] = 0$
$d[i] = 0 \rightarrow x[j] \neq 0$
(for most j's)

A chosen ciphertext attack

- Non-adaptive ciphertext choice $c \equiv -1 \bmod p$ (similar to [YLMH05]):

A chosen ciphertext attack

- Non-adaptive ciphertext choice $c \equiv -1 \bmod p$ (similar to [YLMH05]):
  - RSA: $c = N - 1$

A chosen ciphertext attack

- Non-adaptive ciphertext choice $c \equiv -1 \bmod p$ (similar to [YLMH05]):
  - RSA: $c = N - 1$
  - ElGamal: $c = p - 1$

A chosen ciphertext attack

- Non-adaptive ciphertext choice $c \equiv -1 \bmod p$ (similar to [YLMH05]):
  - RSA: $c = N - 1$
  - ElGamal: $c = p - 1$

- Total #measurements:

Attack type	# of traces	Time	Bandwidth	Cipher

A chosen ciphertext attack

- Non-adaptive ciphertext choice $c \equiv -1 \bmod p$ (similar to [YLMH05]):
  - RSA: $c = N - 1$
  - ElGamal: $c = p - 1$

- Total #measurements:

Attack type	# of traces	Time	Bandwidth	Cipher
Non-adaptive chosen ciphertext	3-15	3 sec	2 MHz	ElGamal, RSA

A chosen ciphertext attack

- Non-adaptive ciphertext choice $c \equiv -1 \bmod p$ (similar to [YLMH05]):
  - RSA: $c = N - 1$
  - ElGamal: $c = p - 1$

- Total #measurements:

Attack type	# of traces	Time	Bandwidth	Cipher
Non-adaptive chosen ciphertext	3-15	3 sec	2 MHz	ElGamal, RSA
Adaptive chosen ciphertext	2048	1 hour	50 kHz	RSA [GST14]

A chosen ciphertext attack

- Non-adaptive ciphertext choice $c \equiv -1 \bmod p$ (similar to [YLMH05]):
  - RSA: $c = N - 1$
  - ElGamal: $c = p - 1$

- Total #measurements:

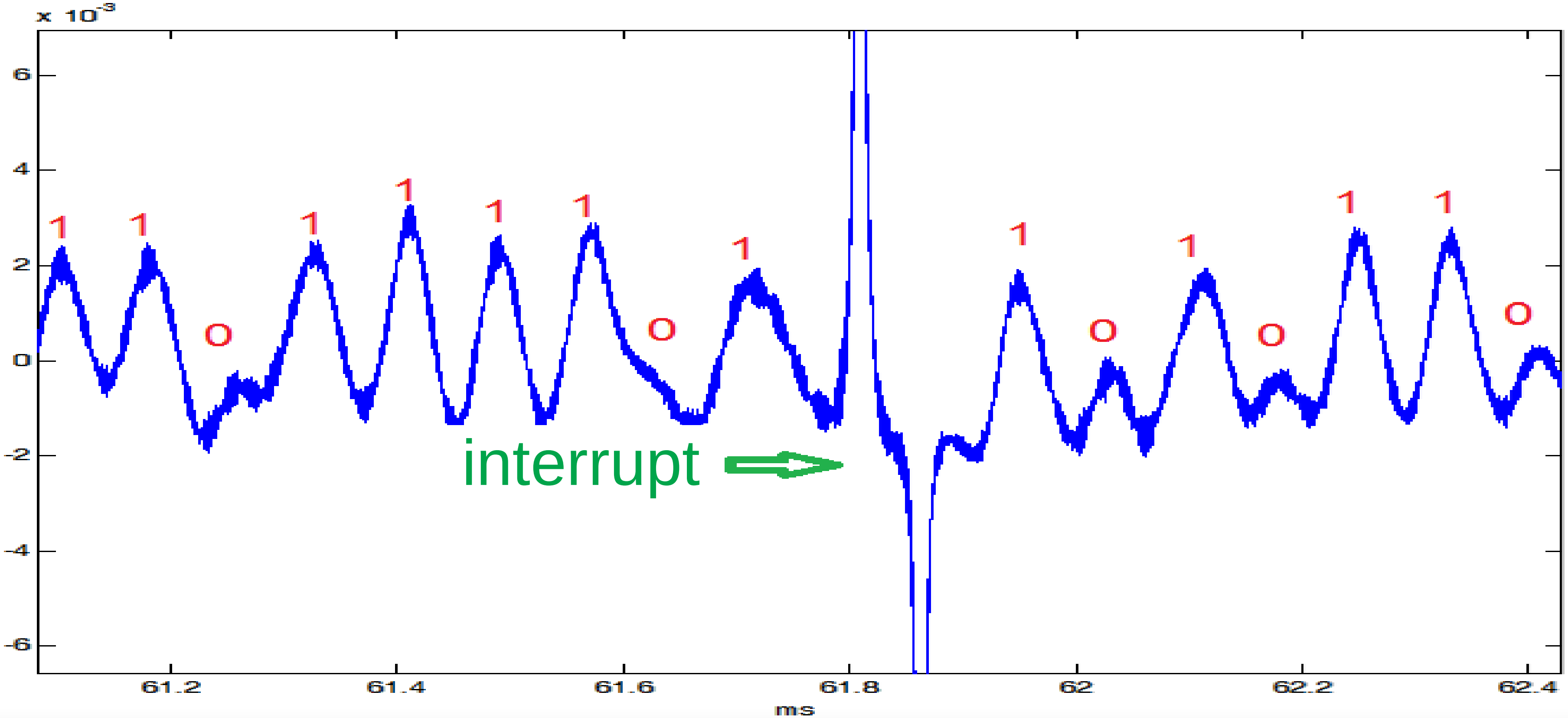
Attack type	# of traces	Time	Bandwidth	Cipher
Non-adaptive chosen ciphertext	3-15	3 sec	2 MHz	ElGamal, RSA
Adaptive chosen ciphertext	2048	1 hour	50 kHz	RSA [GST14]

- Send chosen ciphertexts using Enigmail

Empirical results

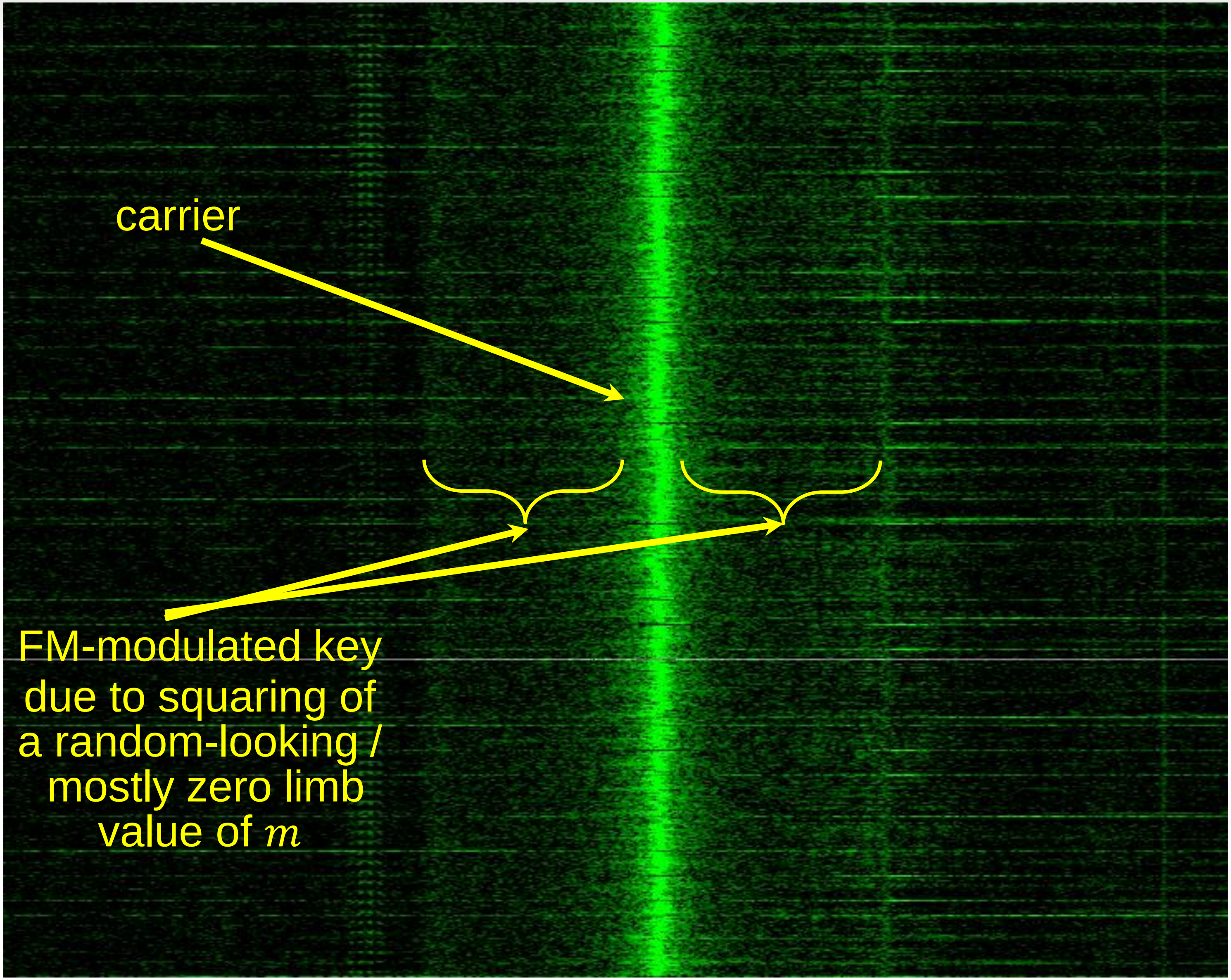
Reading the secret key (non-adaptive attack)

- Acquire trace
- Filter around carrier (1.7 MHz)
- FM demodulation
- Read out bits ("simple ground analysis")

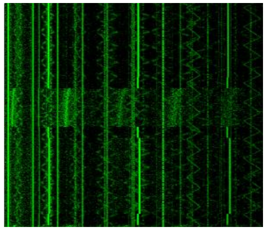


Demo:
key extraction

Reading the secret key (non-adaptive attack)

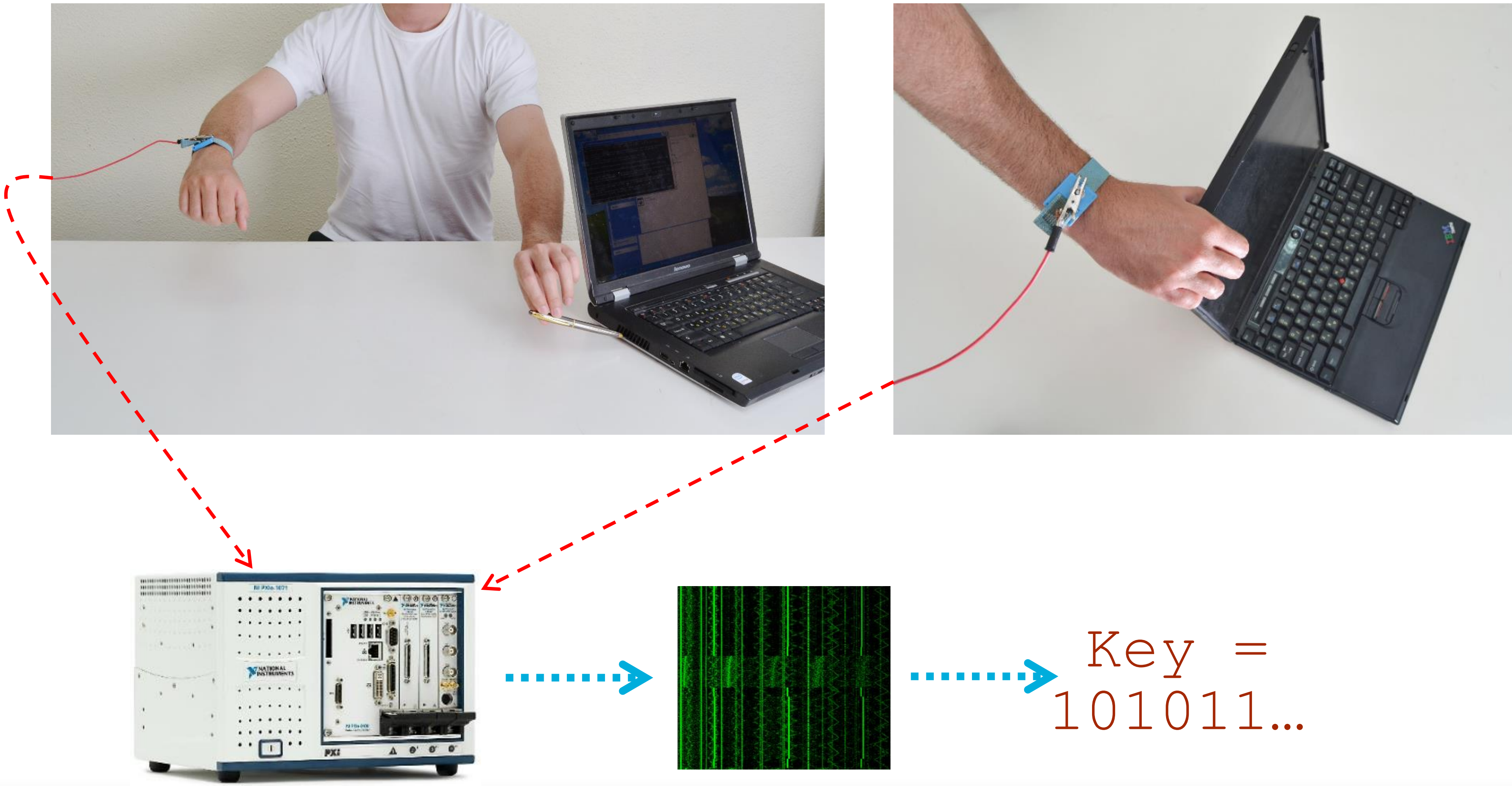


RSA and ElGamal key extraction in a few seconds using direct chassis measurement (non-adaptive attack)



Key = 101011…

RSA and ElGamal key extraction in a few seconds using human touch (non-adaptive attack)



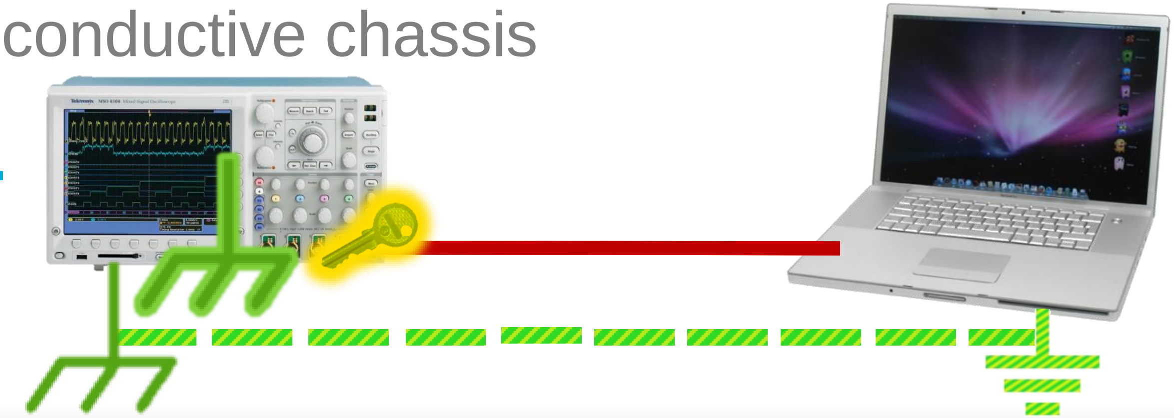
Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis

Key = 101011…

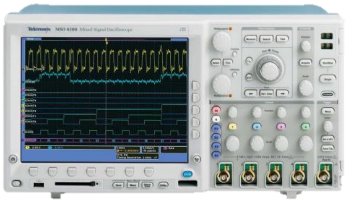


Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis



Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis
connected to	shielded cables

Ground-potential analysis

- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis
connected to	shielded cables

Ground-potential analysis

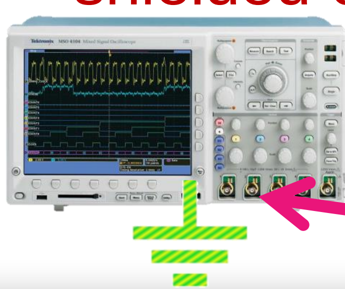
- **Attenuating EMI emanations**
  "Unwanted currents or electromagnetic fields? Dump them to the circuit ground!"
  (Bypass capacitors, RF shields, …)

- Device is grounded, but its "ground" potential fluctuates relative to the mains earth ground.

	Computation
affects	currents and EM fields
dumped to	device ground
connected to	conductive chassis
connected to	shielded cables

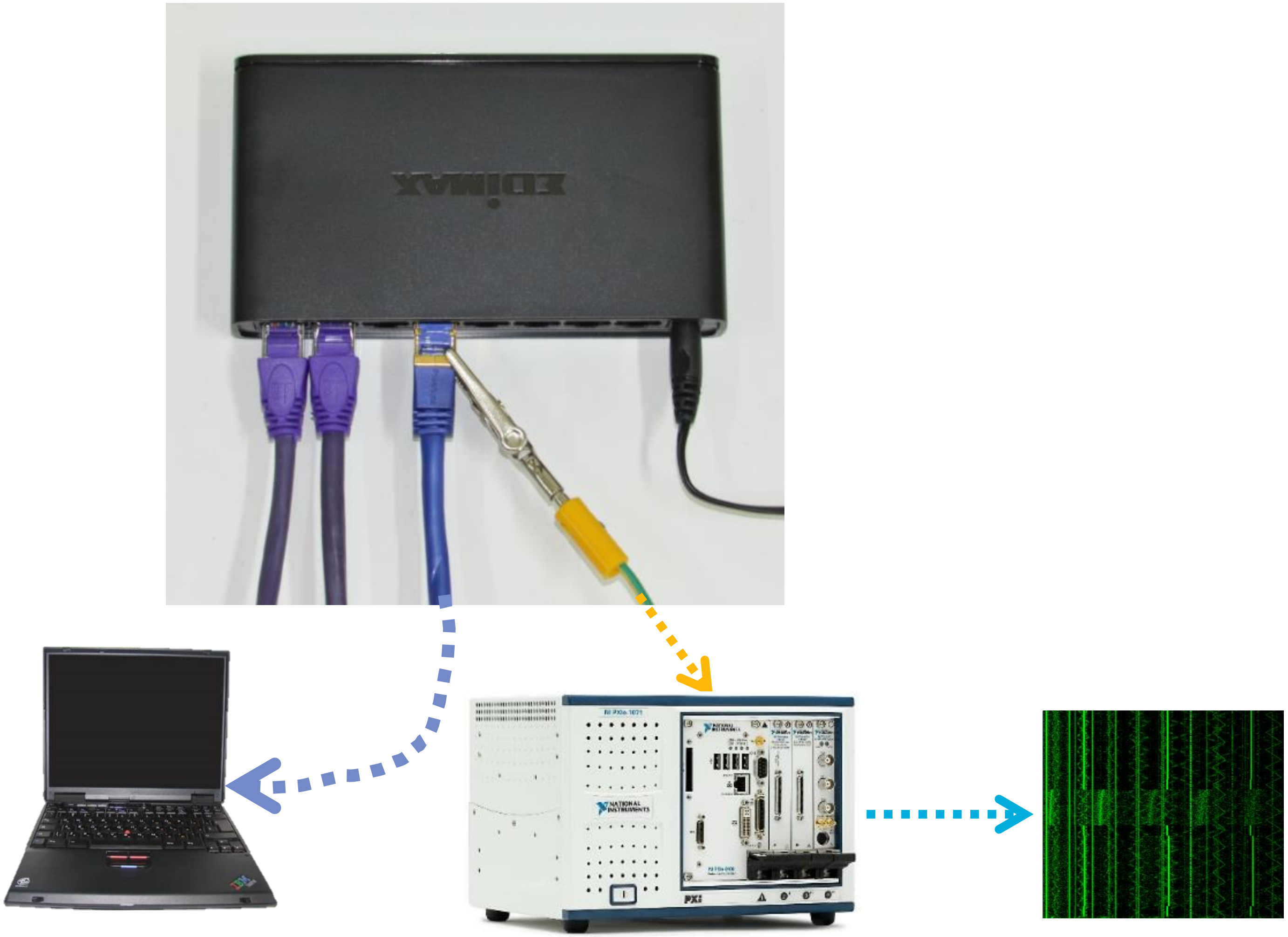
Even when no data, or port is turned off.

Key = 101011…



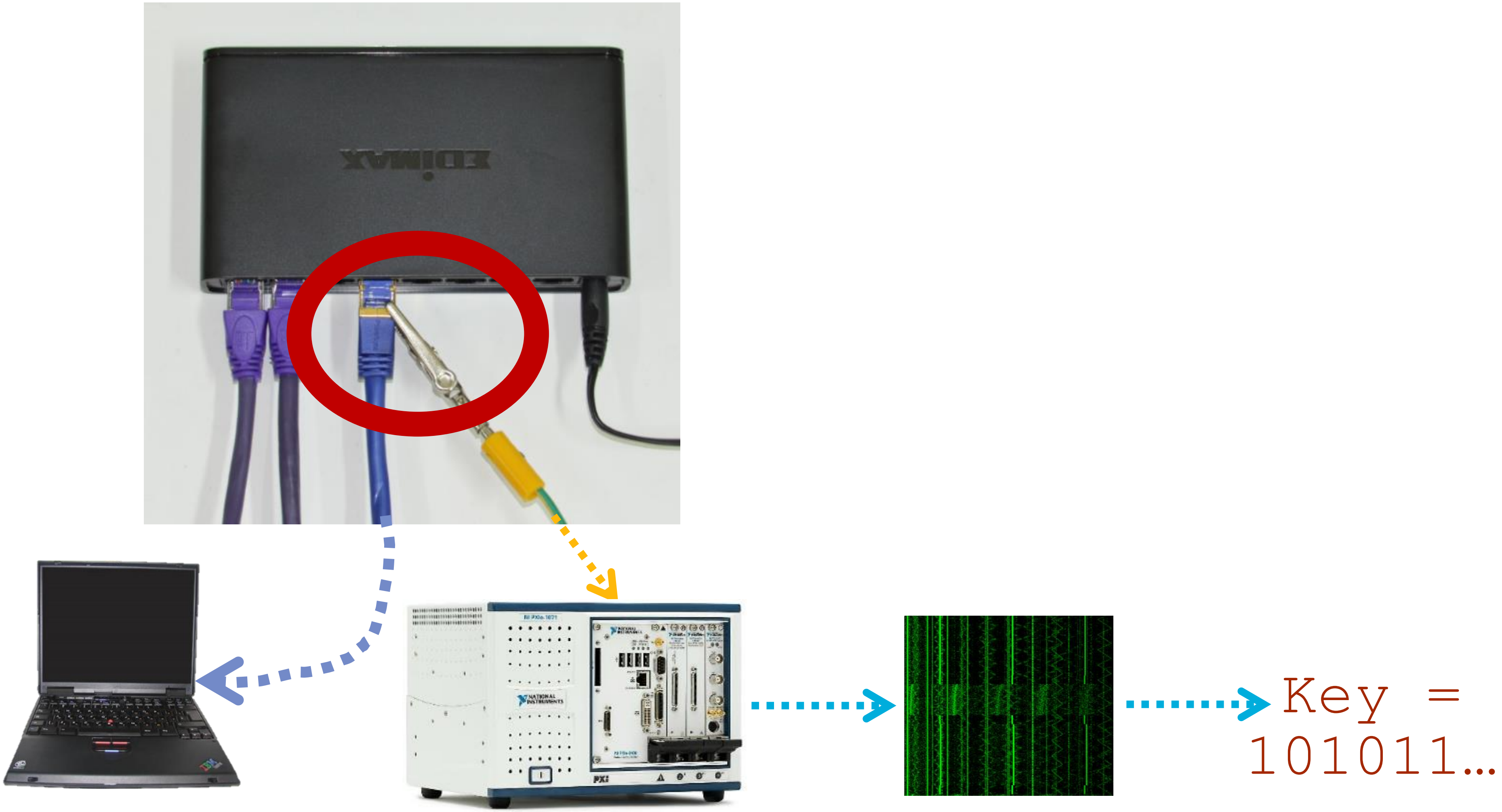
Demo:
key extraction

RSA and ElGamal key extraction in a few seconds using the far end of 10 meter network cable (non-adaptive attack)



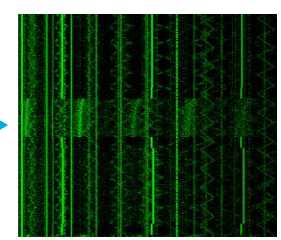
Key = 101011…

RSA and ElGamal key extraction in a few seconds using the far end of 10 meter network cable (non-adaptive attack)



RSA and ElGamal key extraction in a few seconds using the far end of 10 meter network cable (non-adaptive attack)

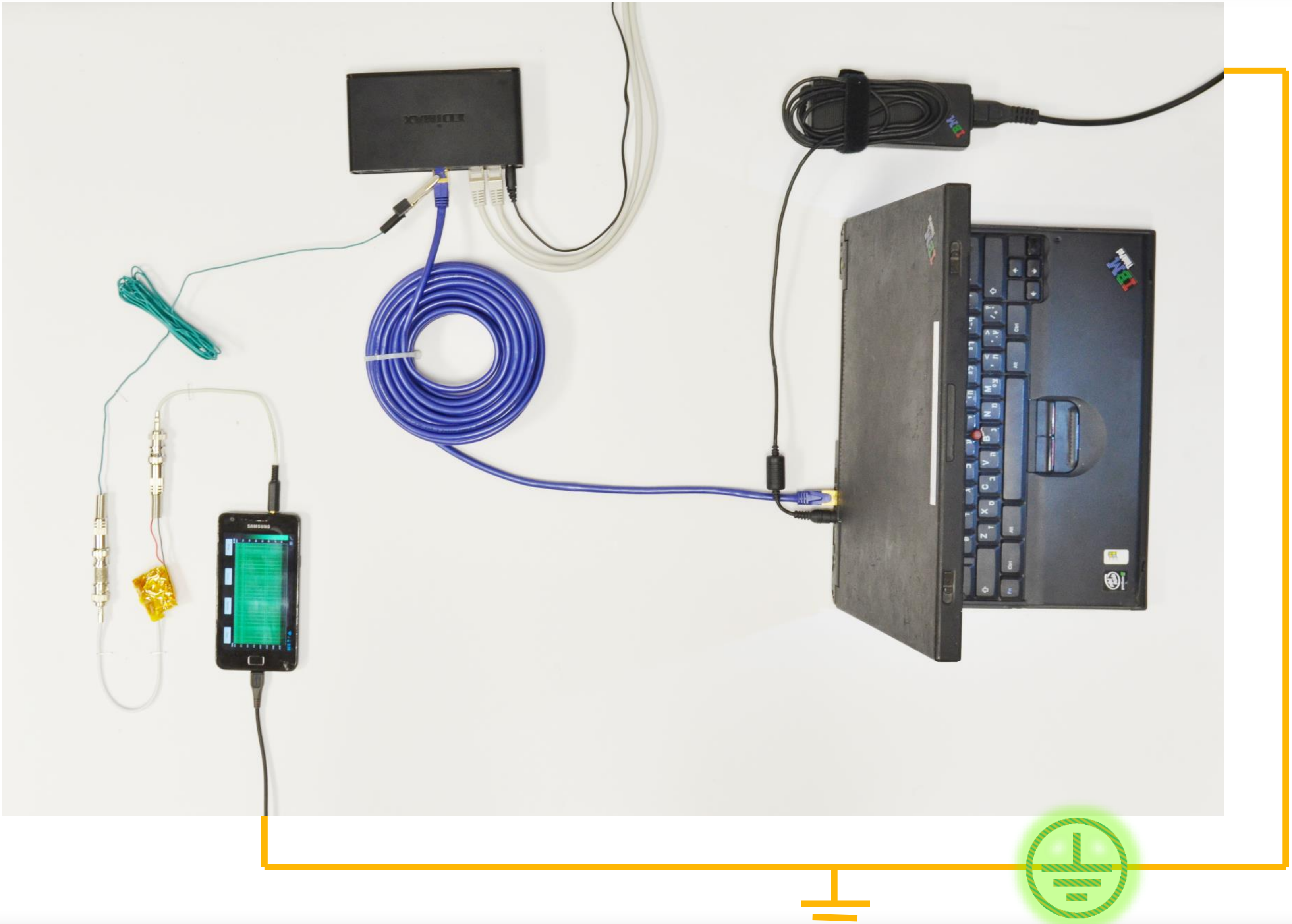
works even if a firewall is present, or port is turned off



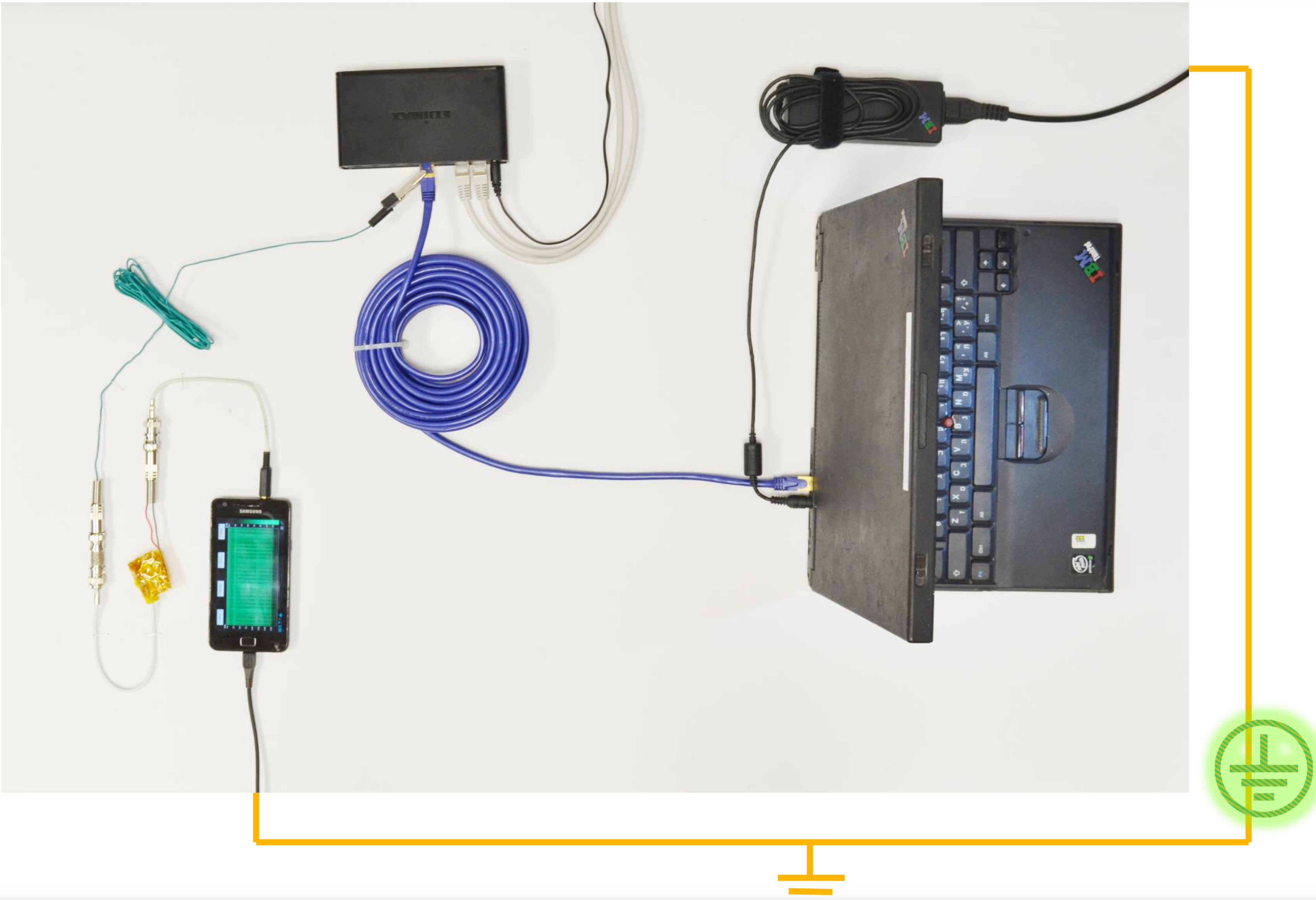
Key = 101011…

Key extraction on far side of Ethernet cable using a mobile phone

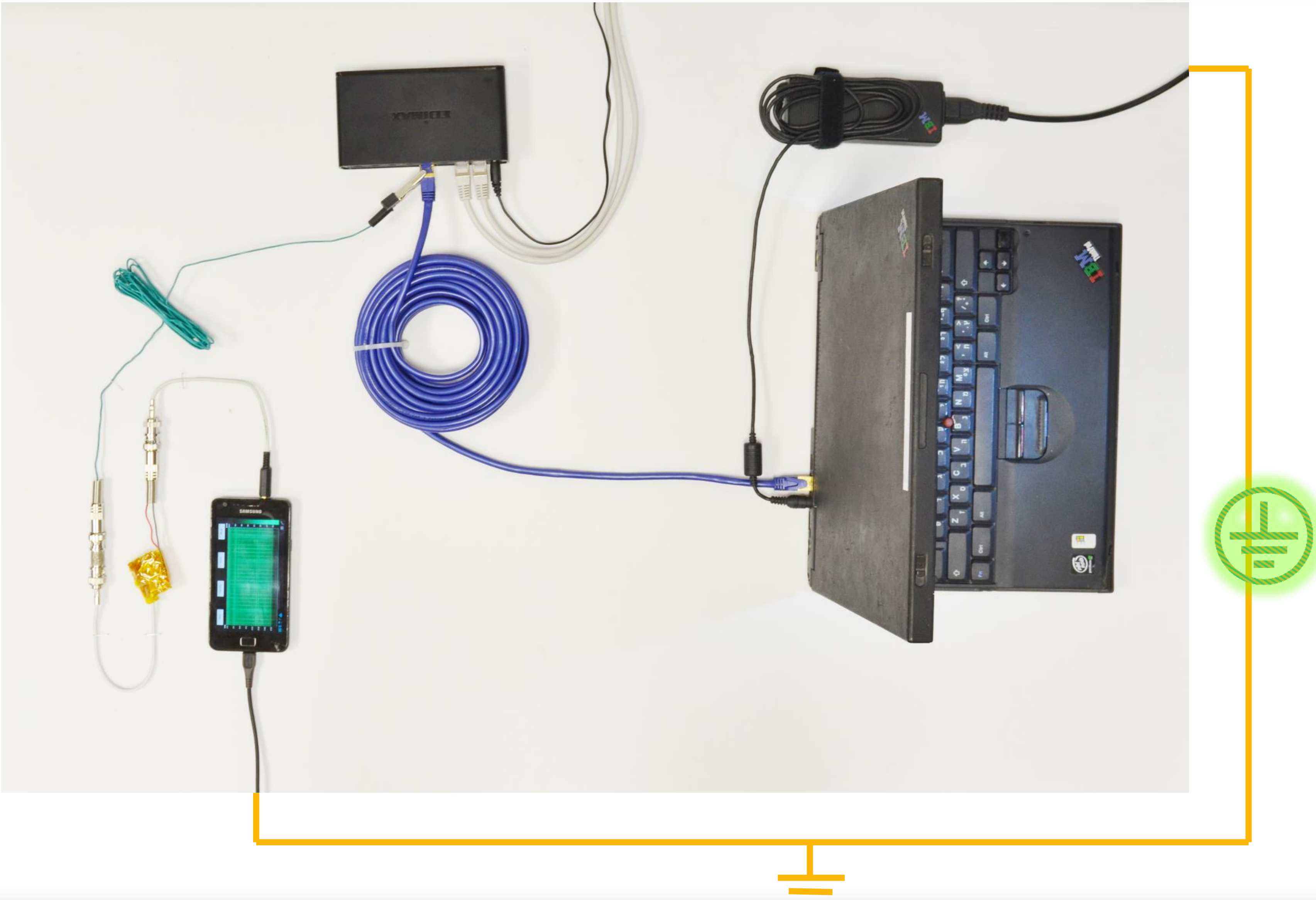
Key extraction on far side of Ethernet cable using a mobile phone



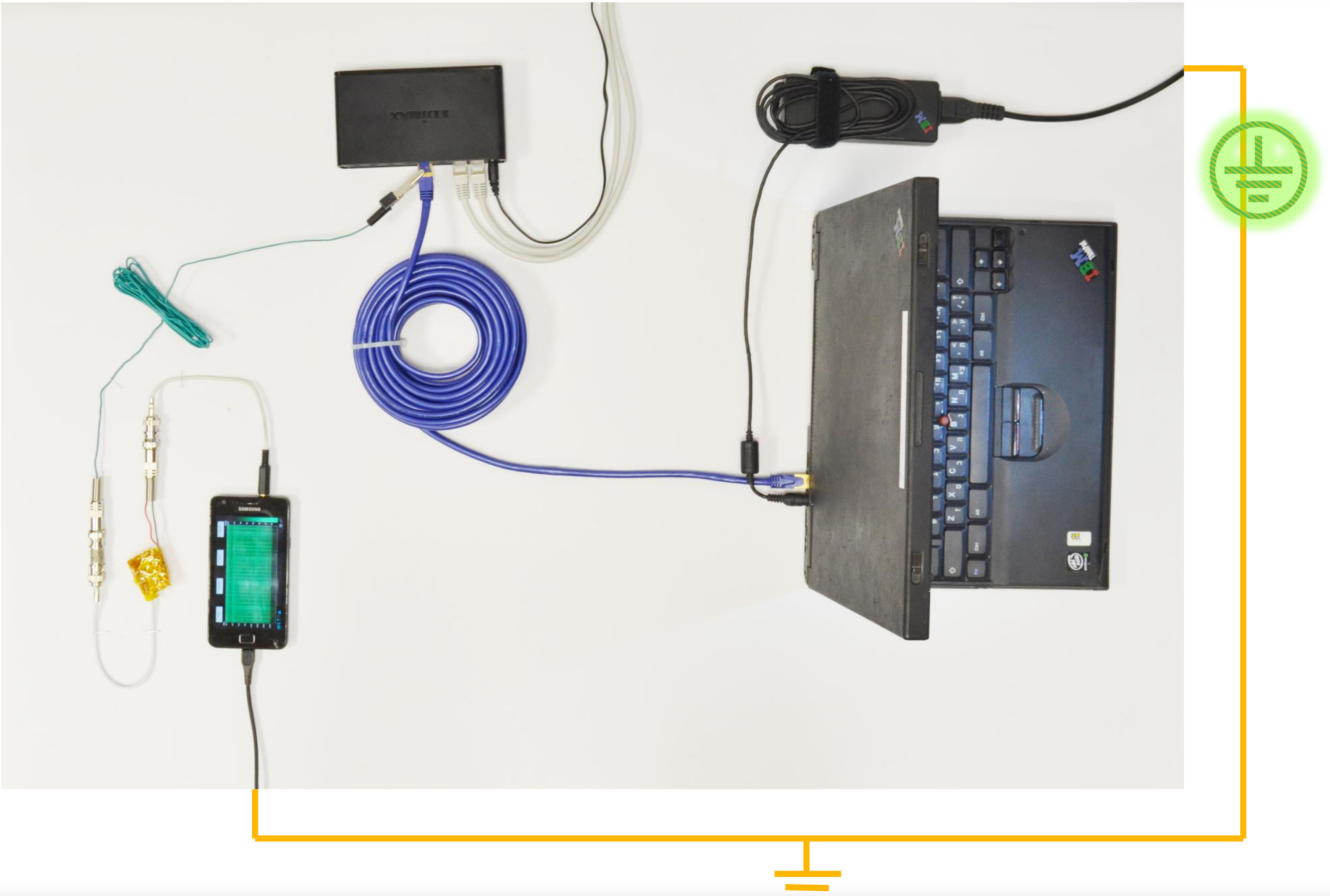
Key extraction on far side of Ethernet cable using a mobile phone



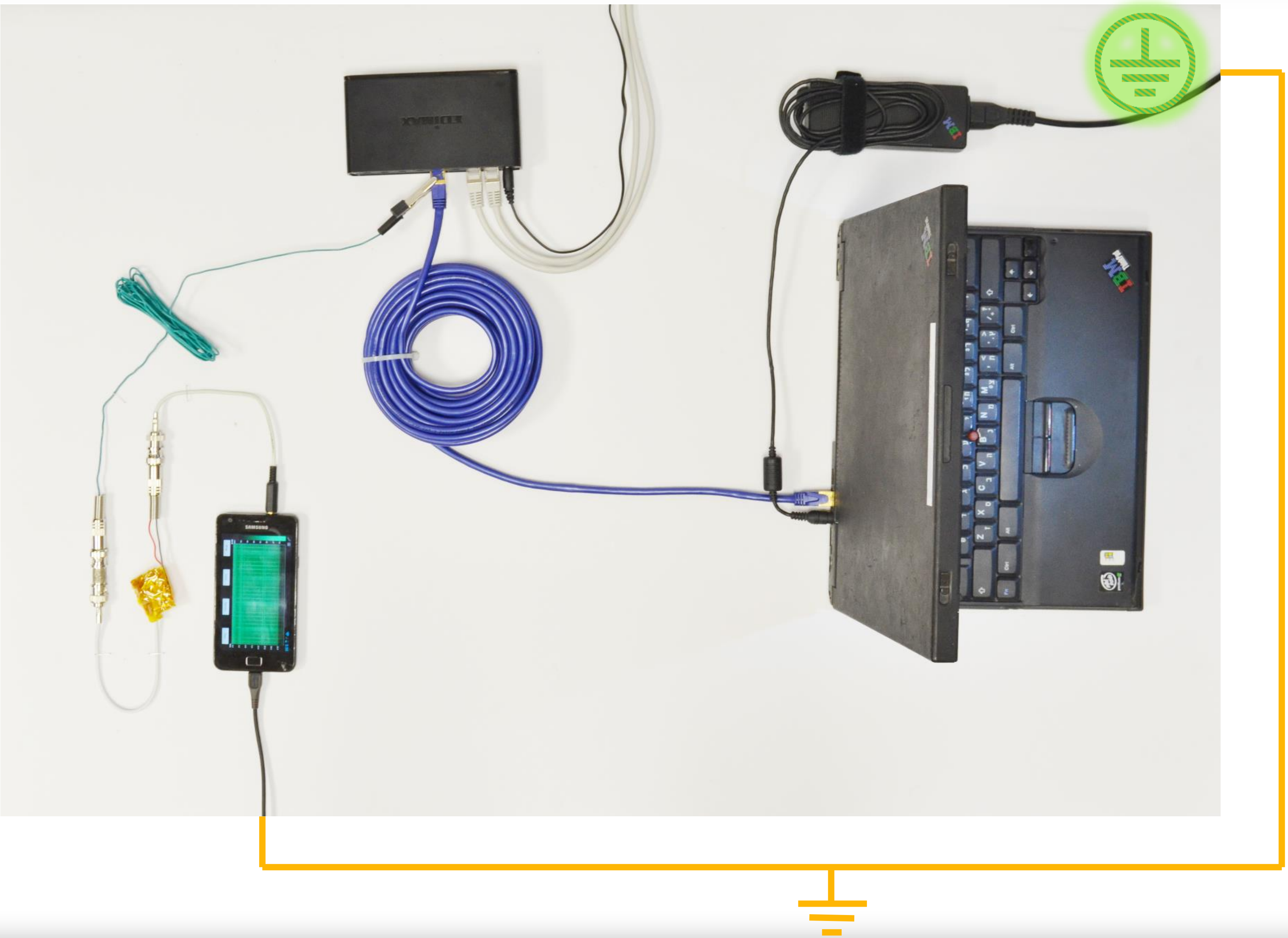
Key extraction on far side of Ethernet cable using a mobile phone



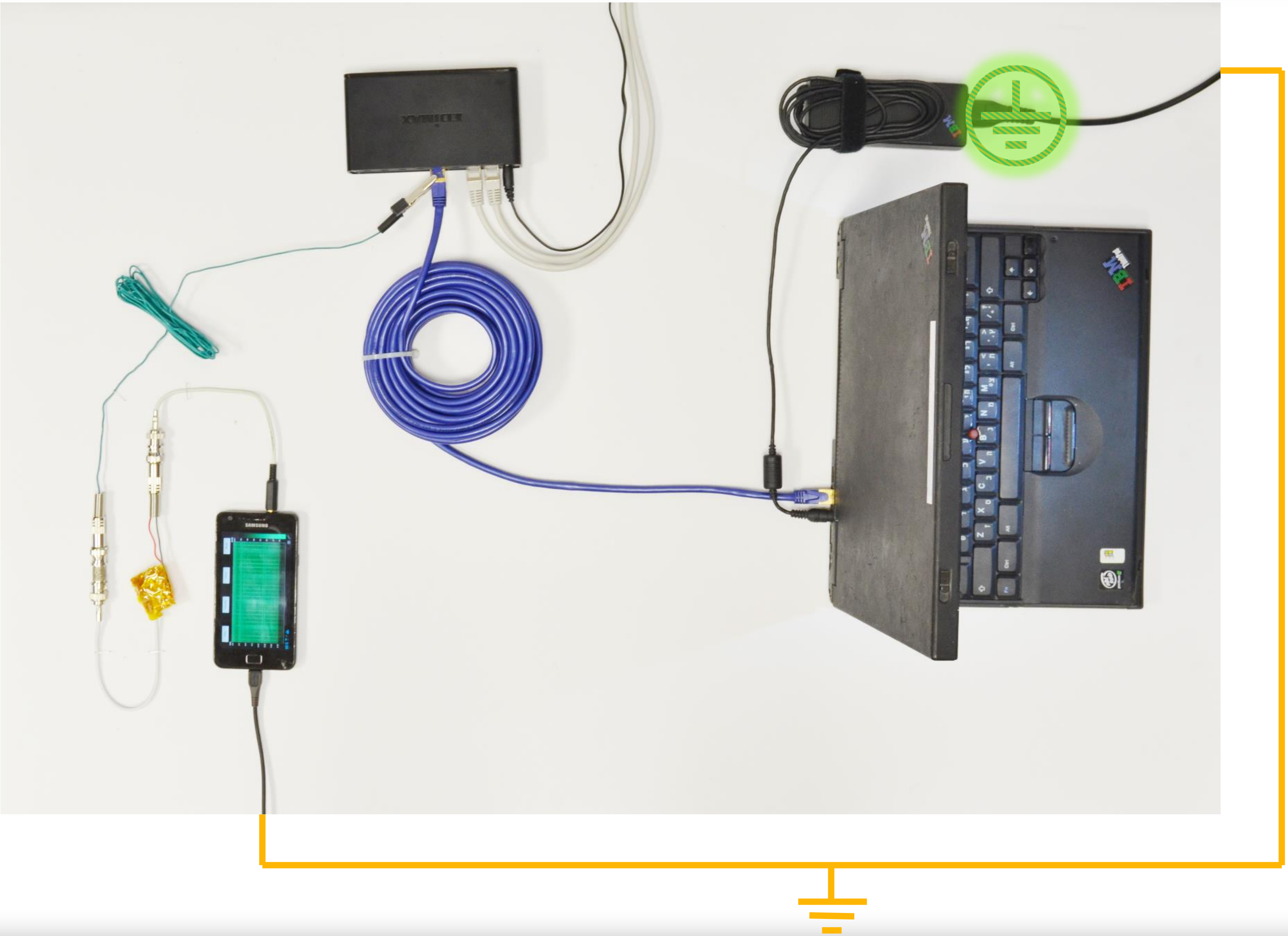
Key extraction on far side of Ethernet cable using a mobile phone



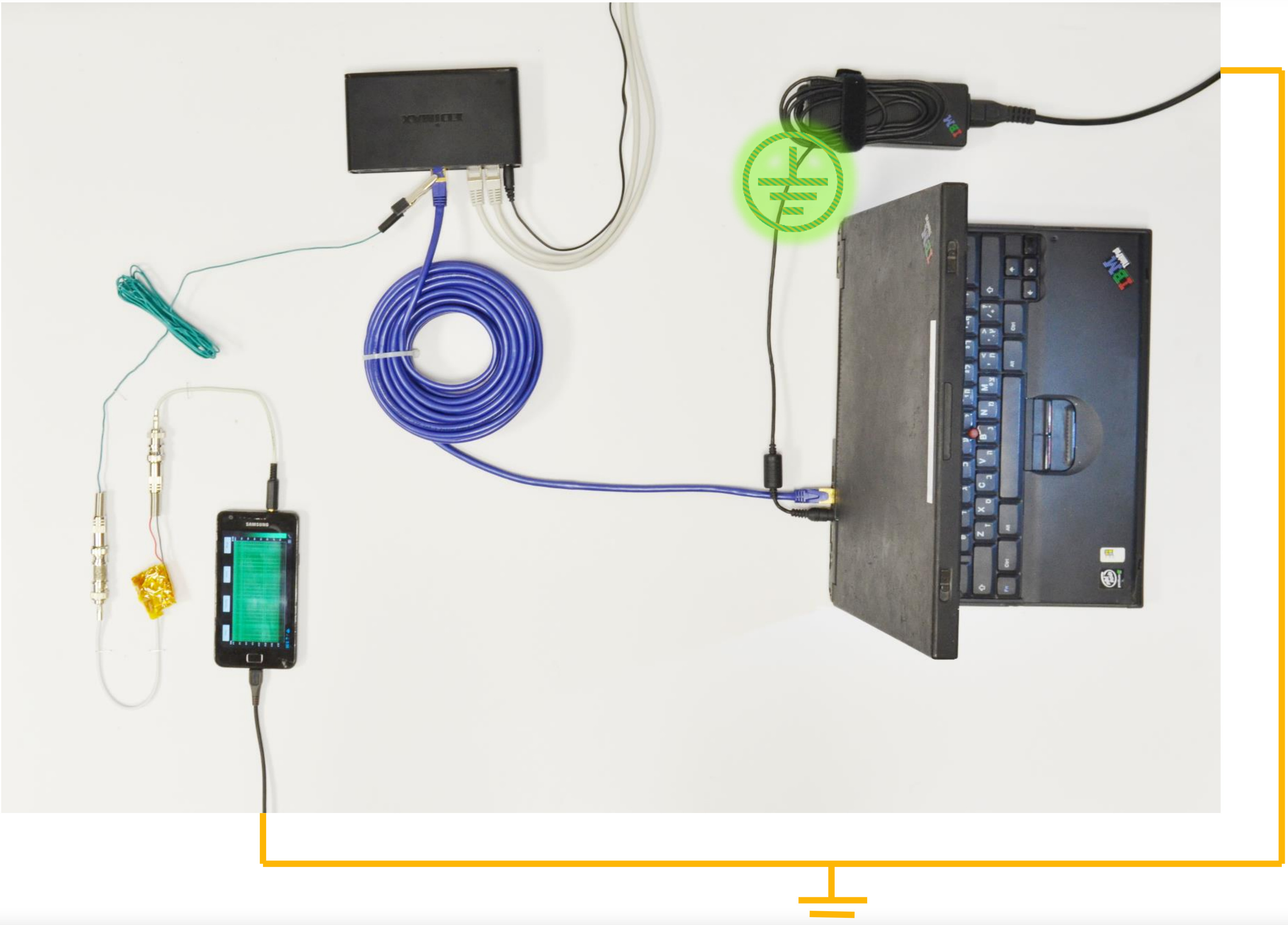
Key extraction on far side of Ethernet cable using a mobile phone



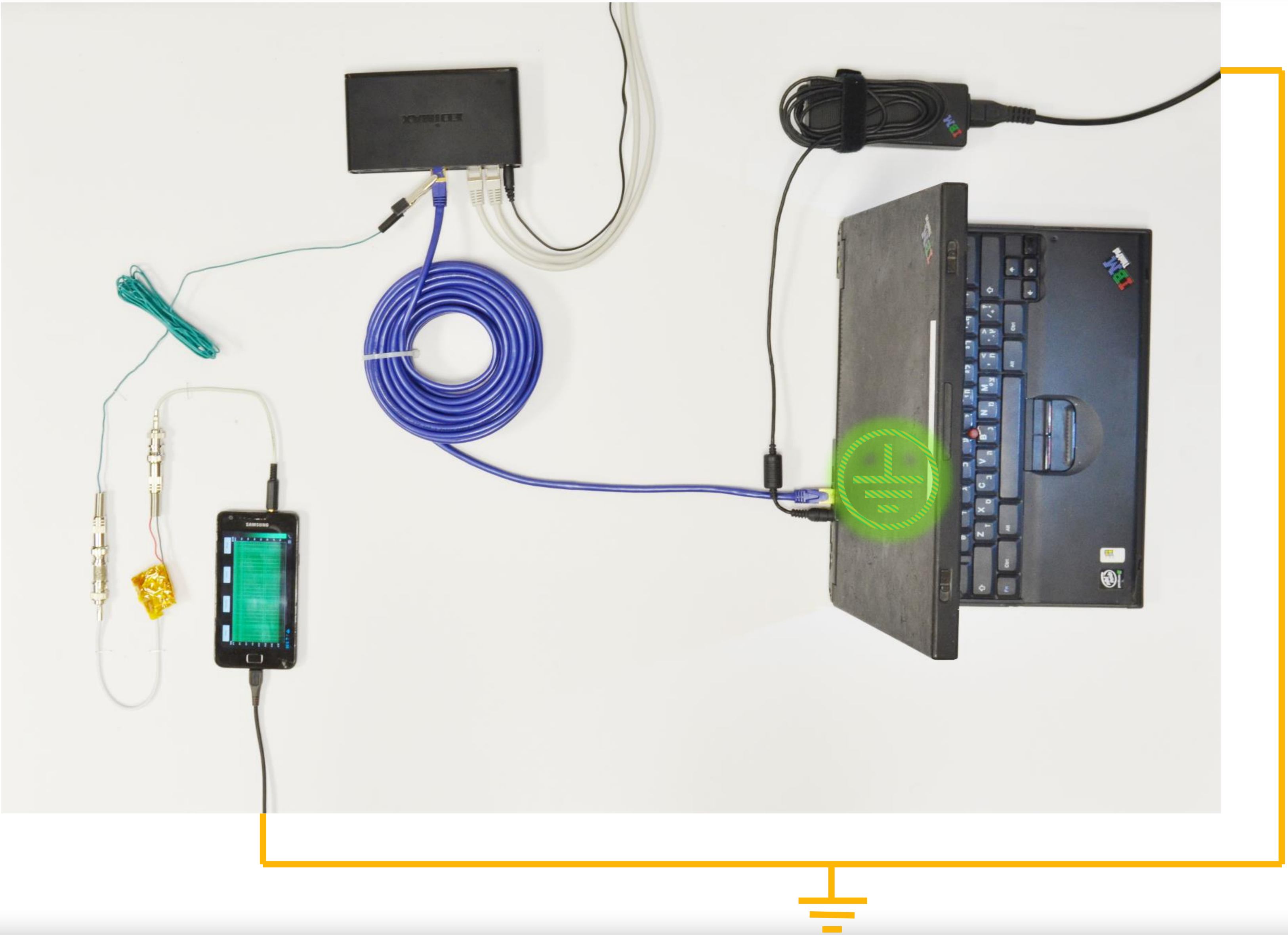
Key extraction on far side of Ethernet cable using a mobile phone



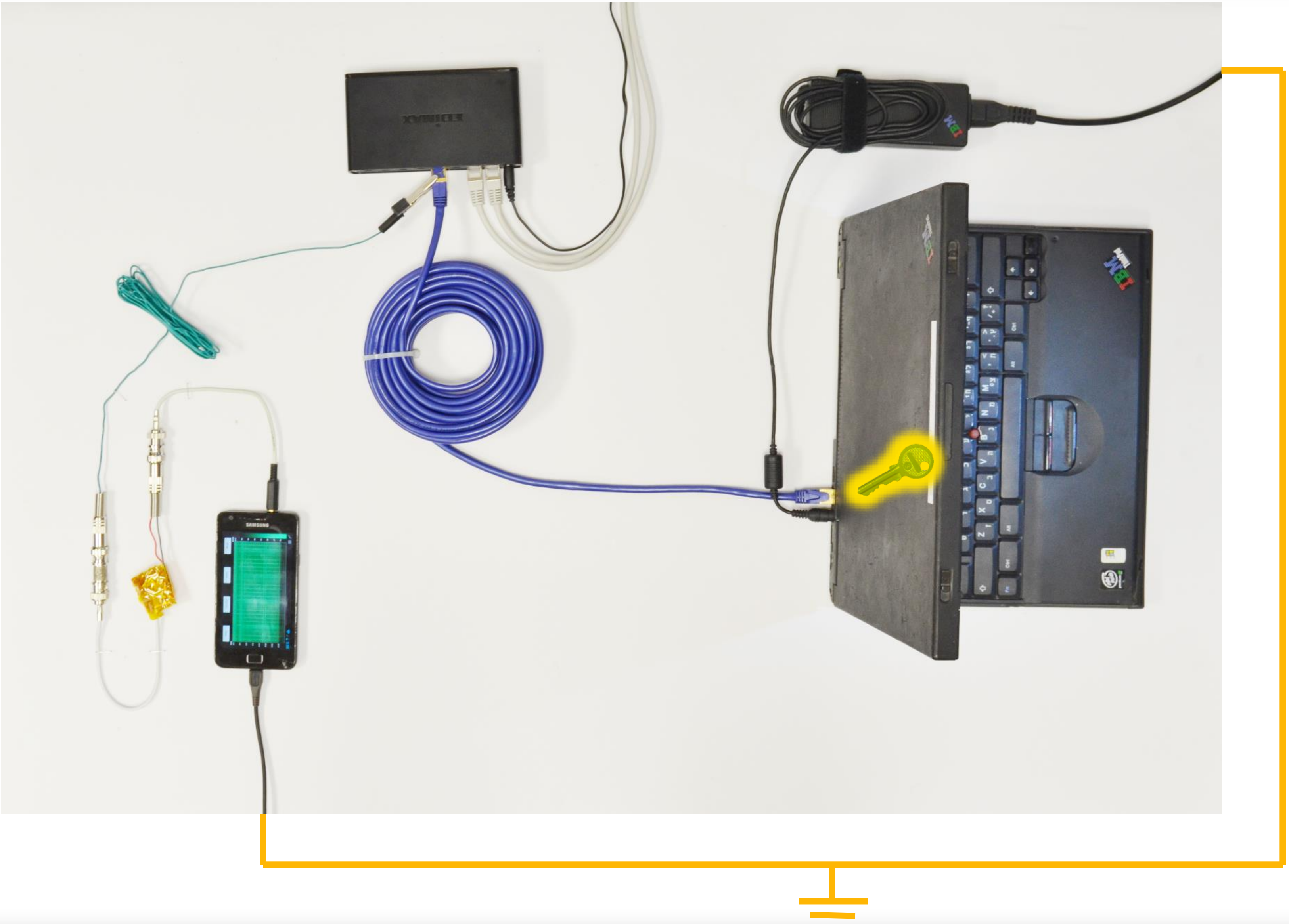
Key extraction on far side of Ethernet cable using a mobile phone



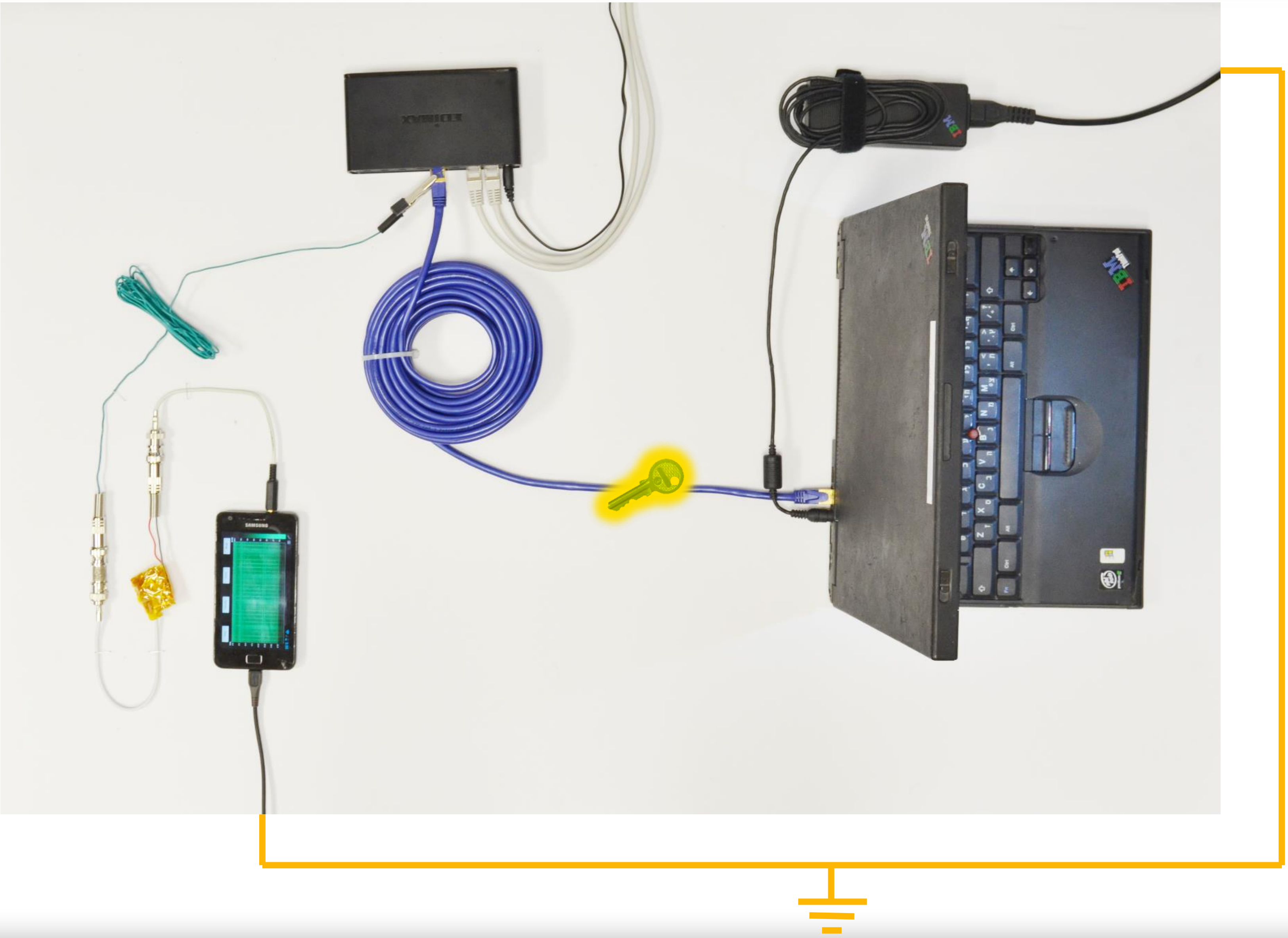
Key extraction on far side of Ethernet cable using a mobile phone



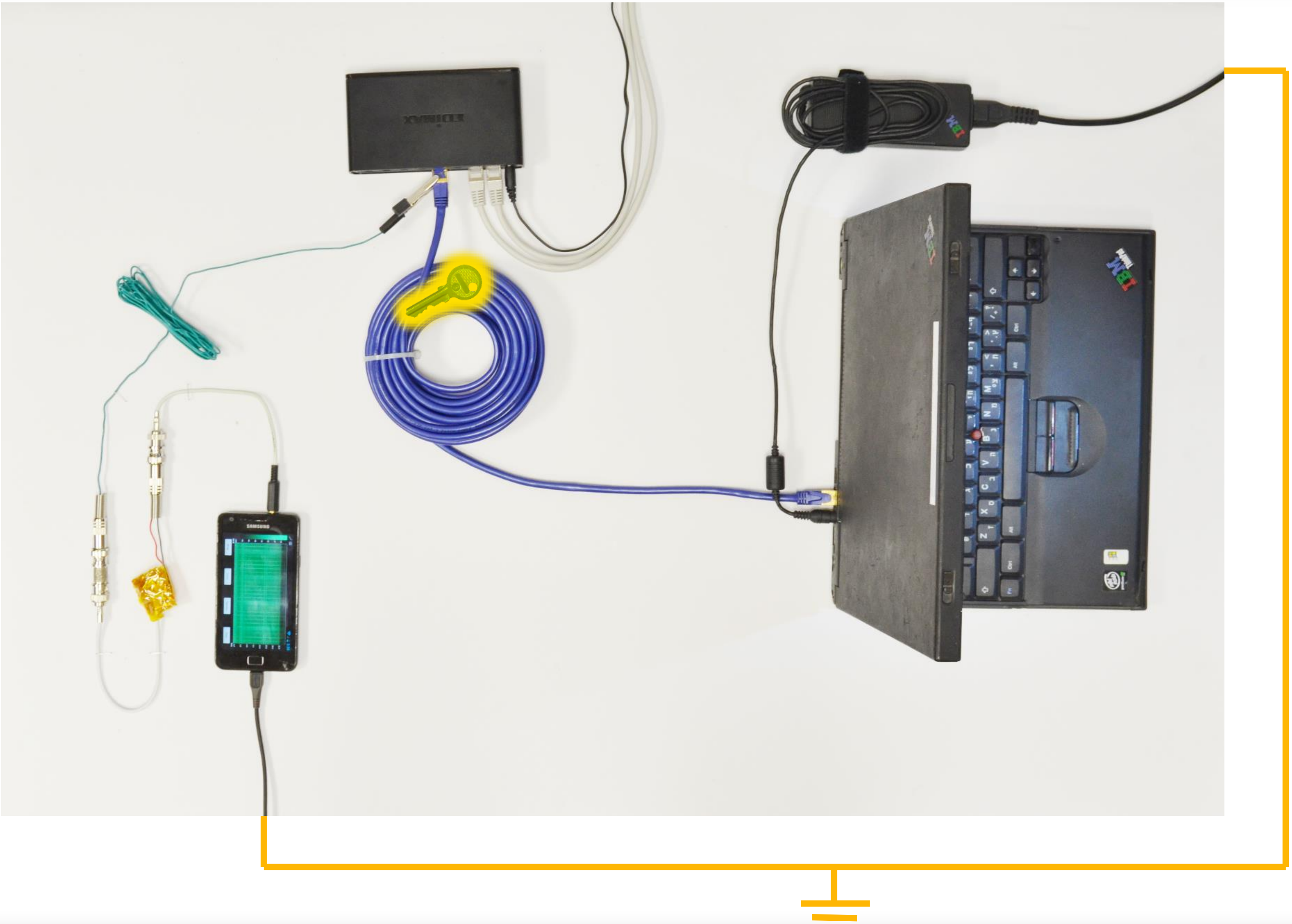
Key extraction on far side of Ethernet cable using a mobile phone



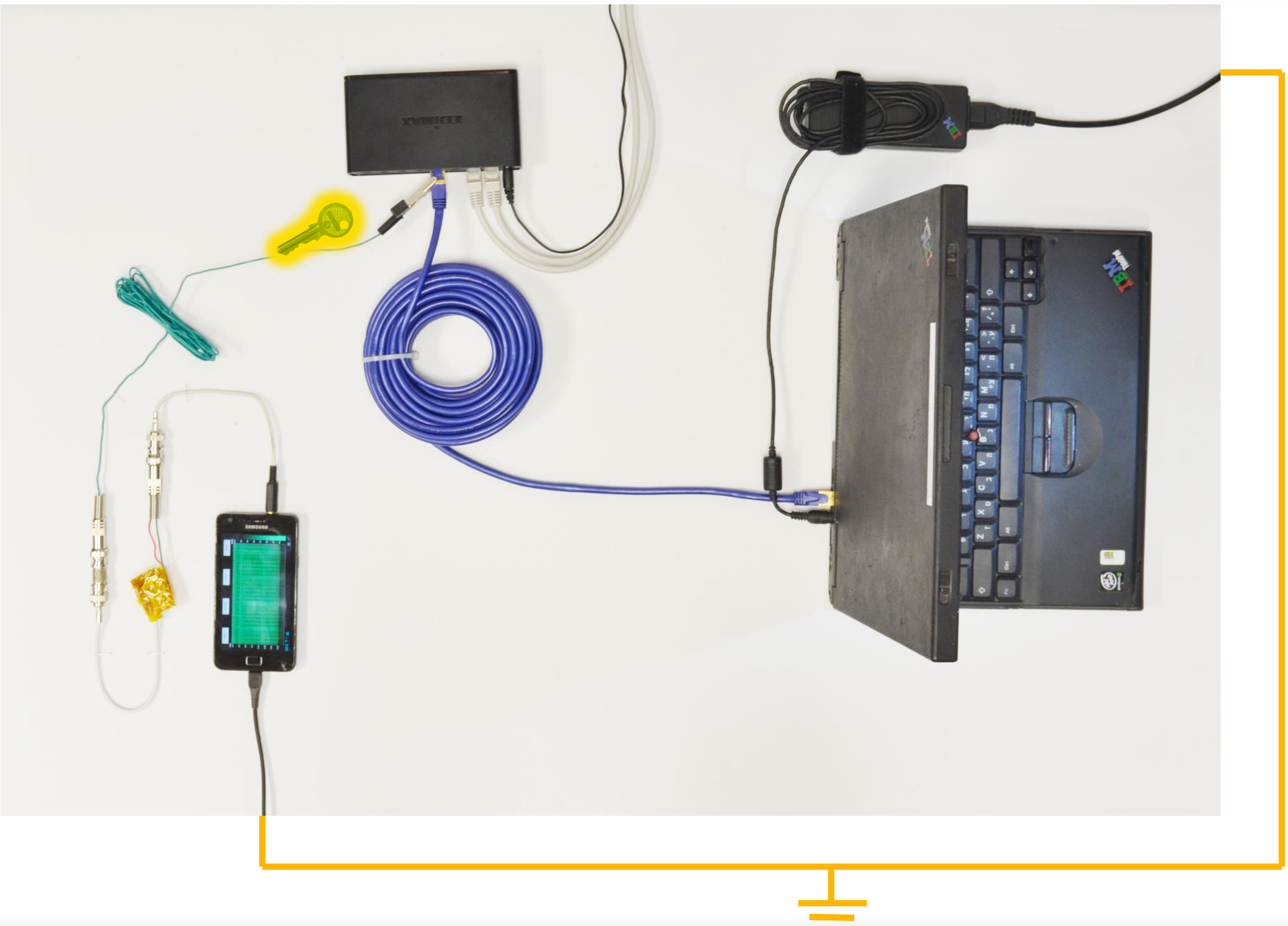
Key extraction on far side of Ethernet cable using a mobile phone



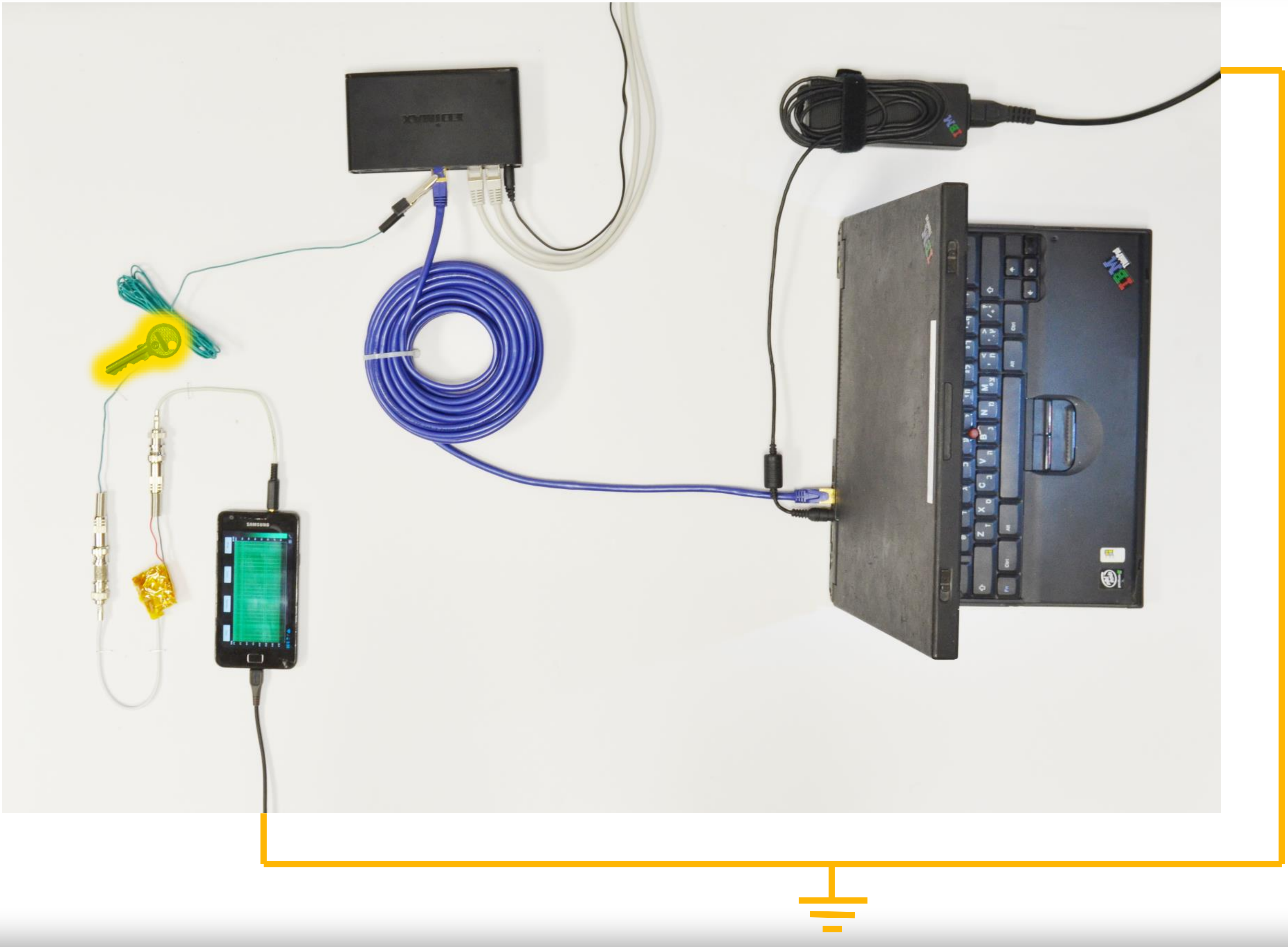
Key extraction on far side of Ethernet cable using a mobile phone



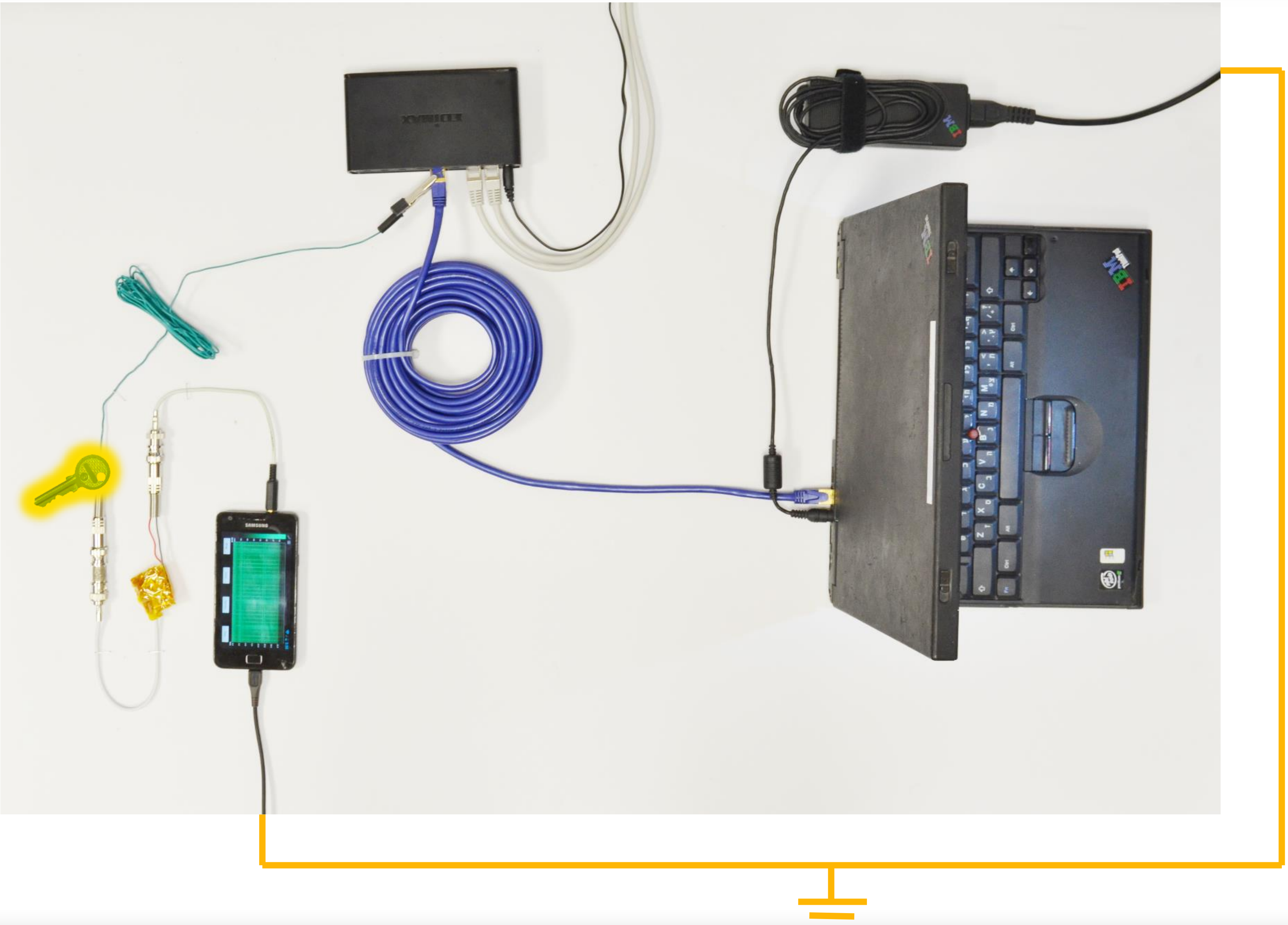
Key extraction on far side of Ethernet cable using a mobile phone



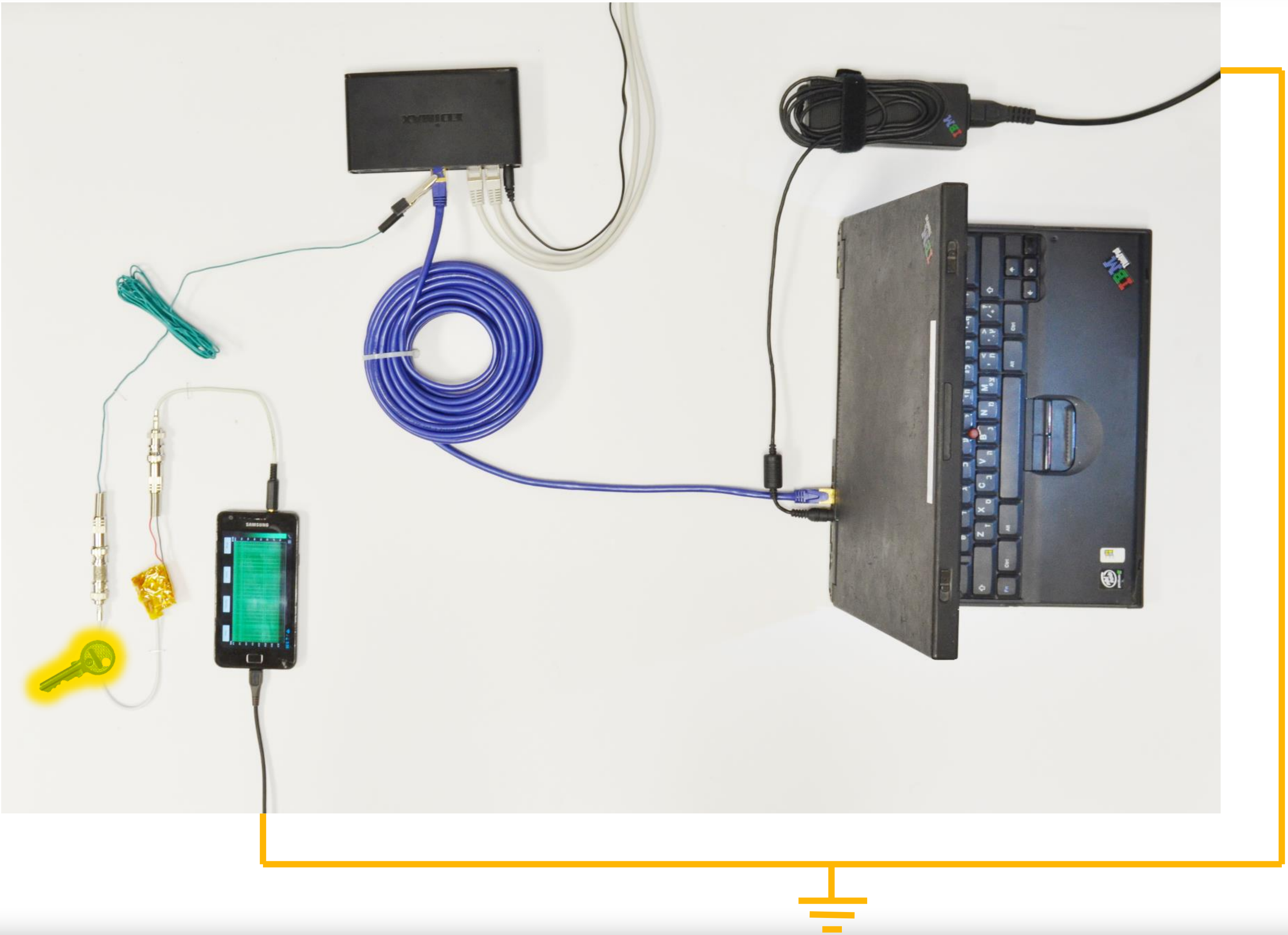
Key extraction on far side of Ethernet cable using a mobile phone



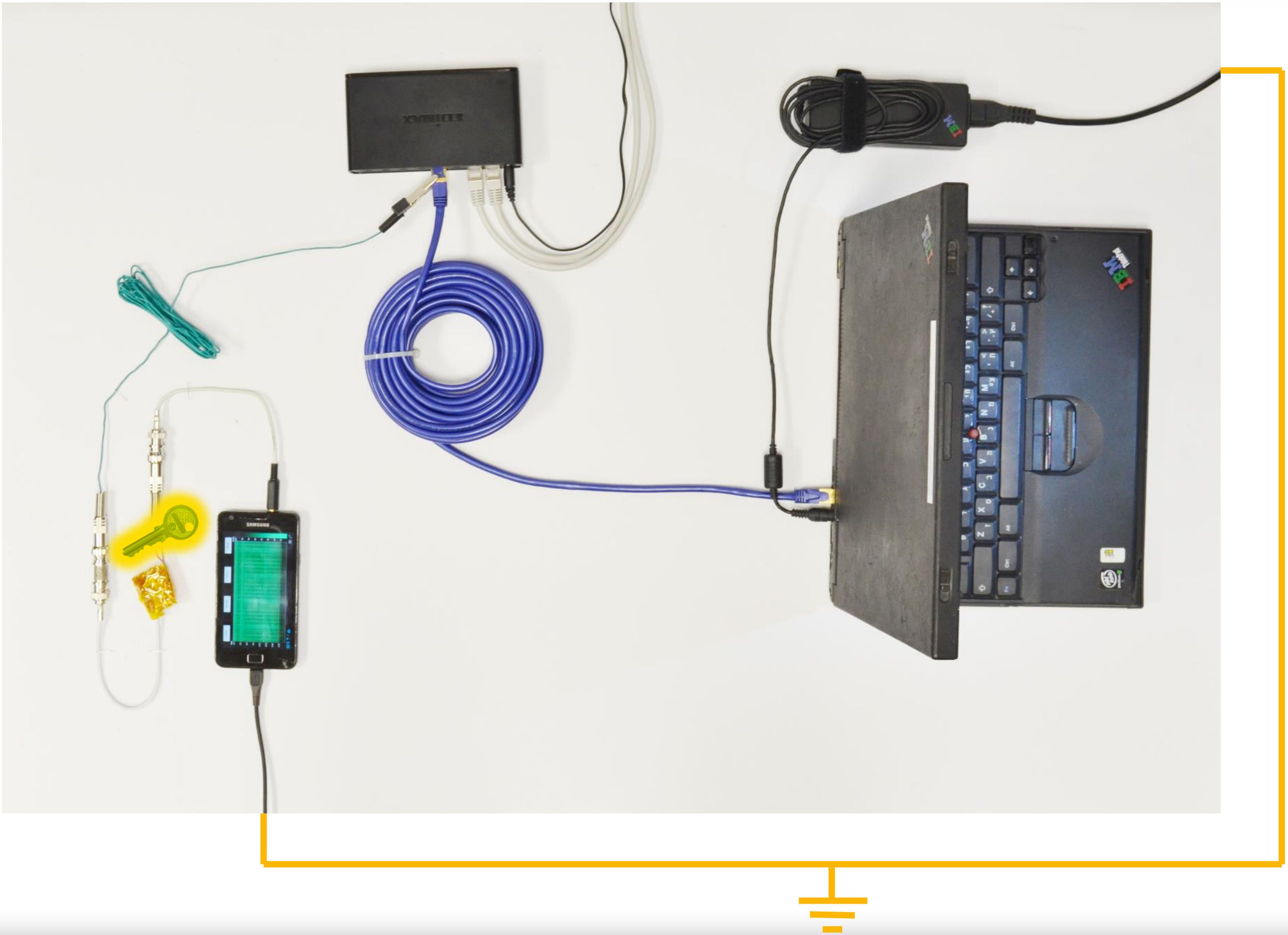
Key extraction on far side of Ethernet cable using a mobile phone



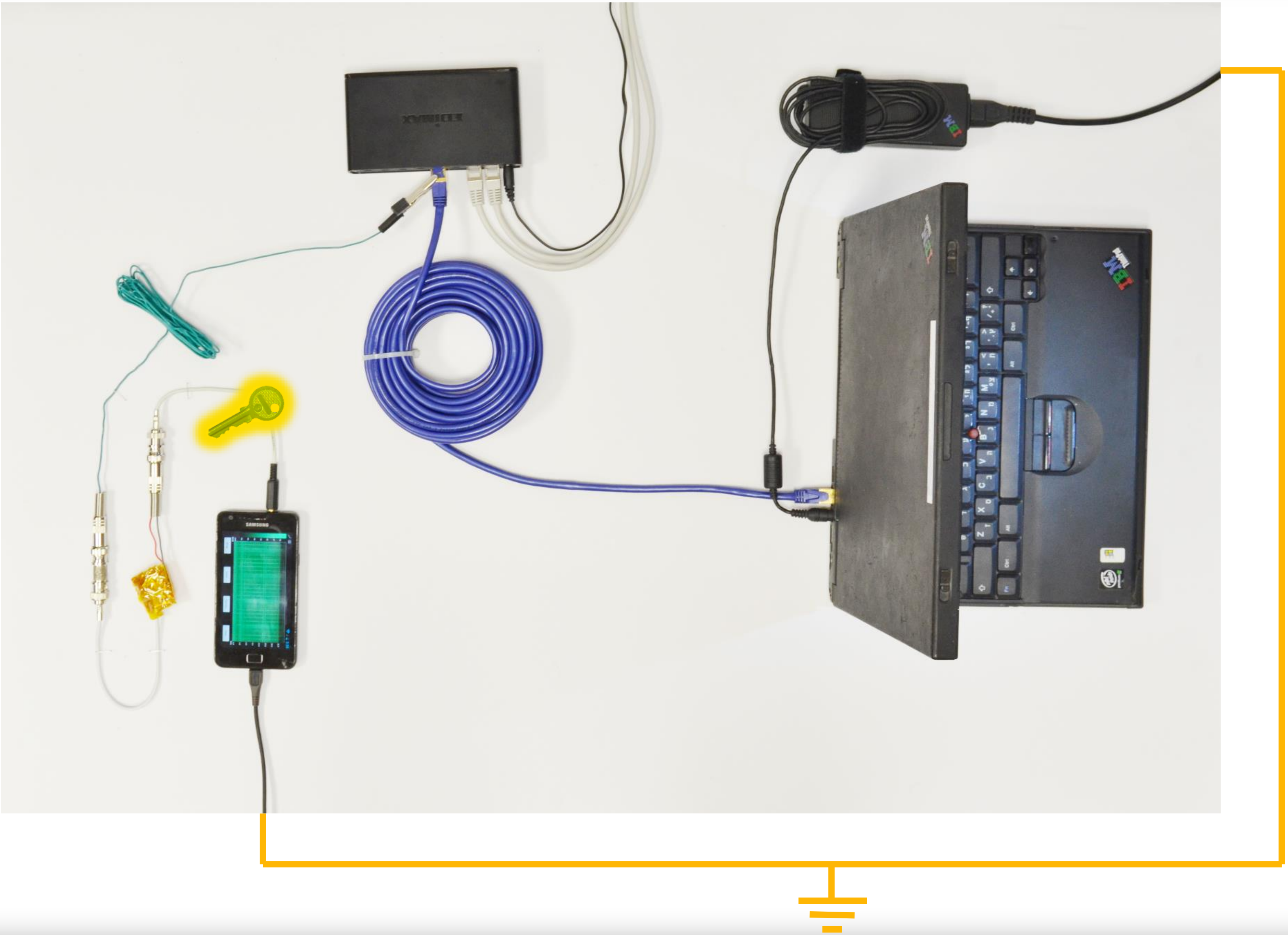
Key extraction on far side of Ethernet cable using a mobile phone



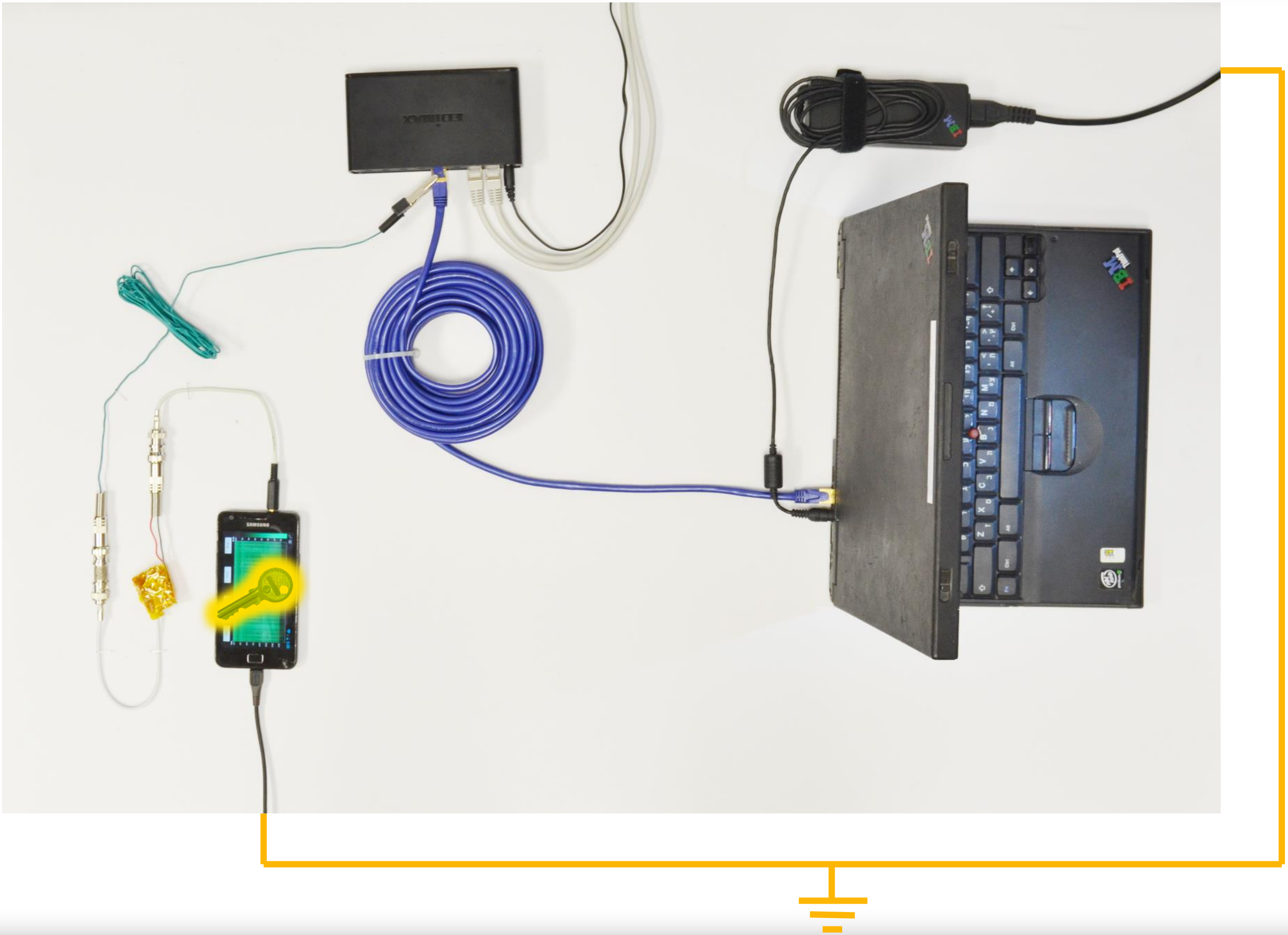
Key extraction on far side of Ethernet cable using a mobile phone



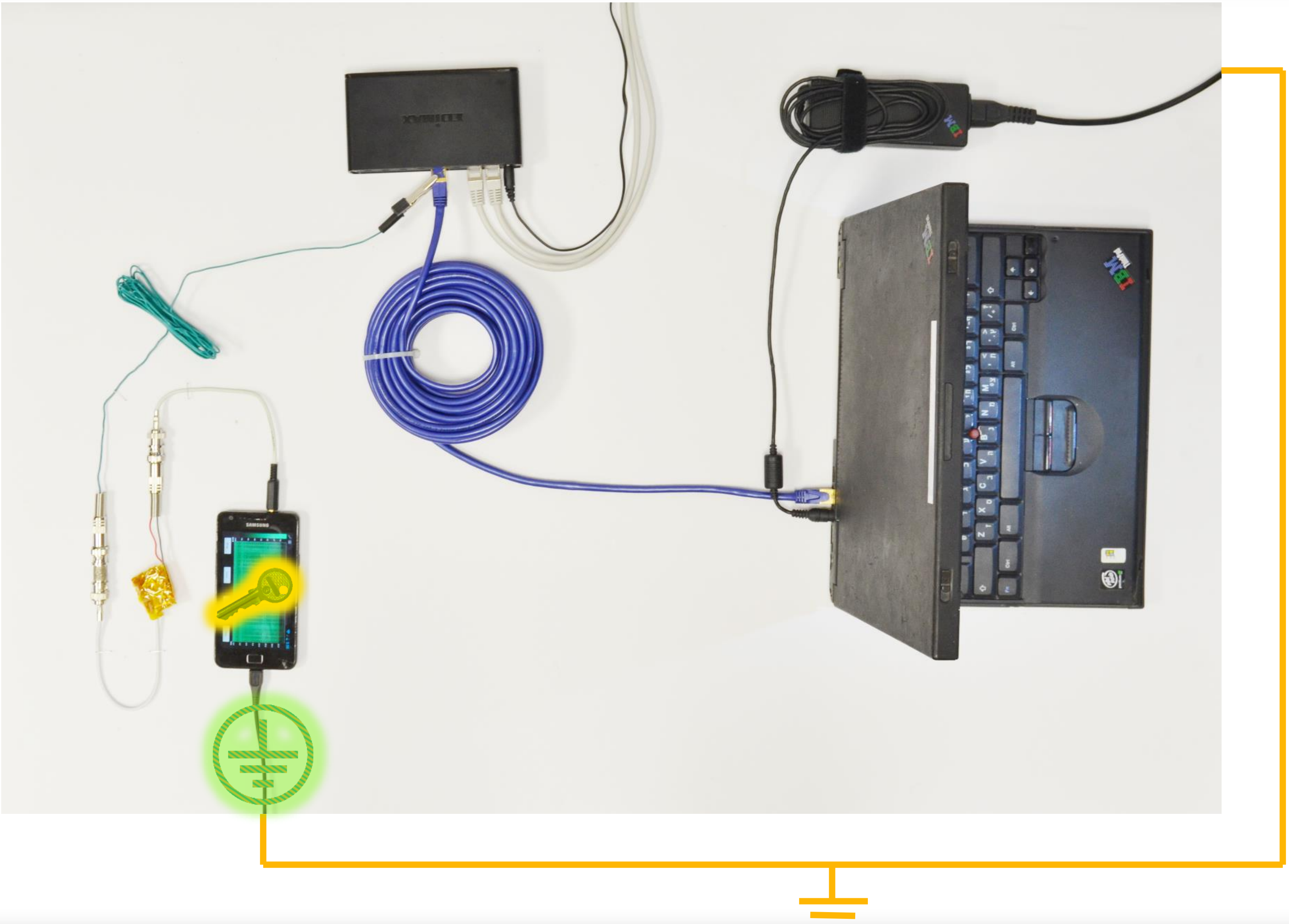
Key extraction on far side of Ethernet cable using a mobile phone



Key extraction on far side of Ethernet cable using a mobile phone



Key extraction on far side of Ethernet cable using a mobile phone



Countermeasures

Ineffective countermeasures:

1. Add analog noise
2. Parallel software load

Countermeasures

Ineffective countermeasures:

1. Add analog noise
2. Parallel software load

Countermeasures

Ineffective countermeasures:

1. Add analog noise
2. Parallel software load

Main problem: decryption of adversarial inputs

Countermeasures

Ineffective countermeasures:

1. Add analog noise
2. Parallel software load

Main problem: decryption of adversarial inputs

Solution: ciphertext randomization use equivalent but random-looking ciphertexts

Countermeasures

Ineffective countermeasures:

1. Add analog noise
2. Parallel software load

Main problem: decryption of adversarial inputs

Solution: ciphertext randomization use equivalent but random-looking ciphertexts

- Negligible slowdown for RSA

Countermeasures

Ineffective countermeasures:

1. Add analog noise
2. Parallel software load

Main problem: decryption of adversarial inputs

Solution: ciphertext randomization use equivalent but random-looking ciphertexts

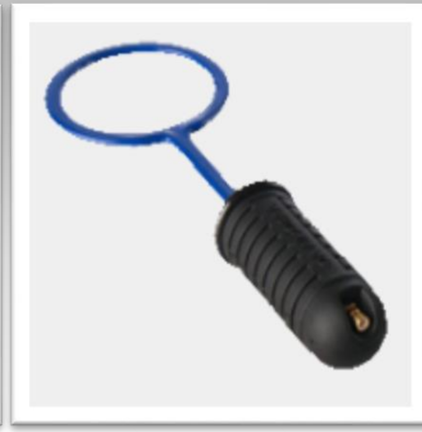
- Negligible slowdown for RSA
- x2 slowdown for ElGamal



Thanks!

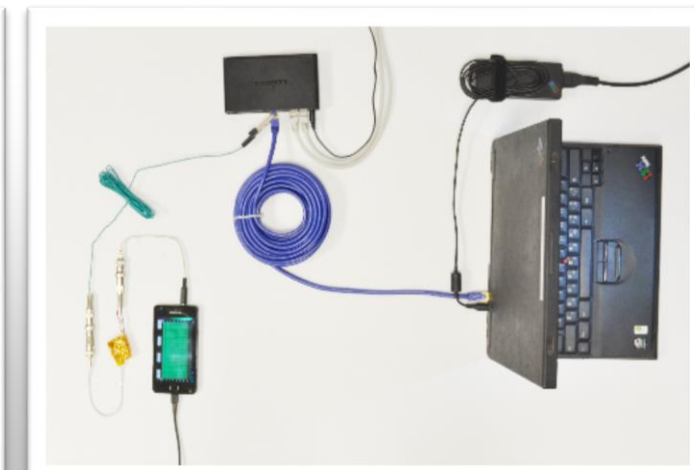
cs.tau.ac.il/~tromer/handsoff

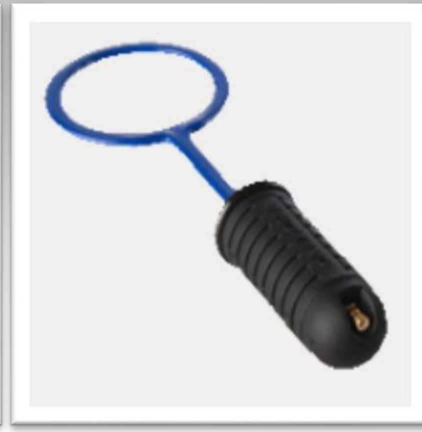




Thanks!

cs.tau.ac.il/~tromer/handsoff




Thanks!

`cs.tau.ac.il/~tromer/handsoff`